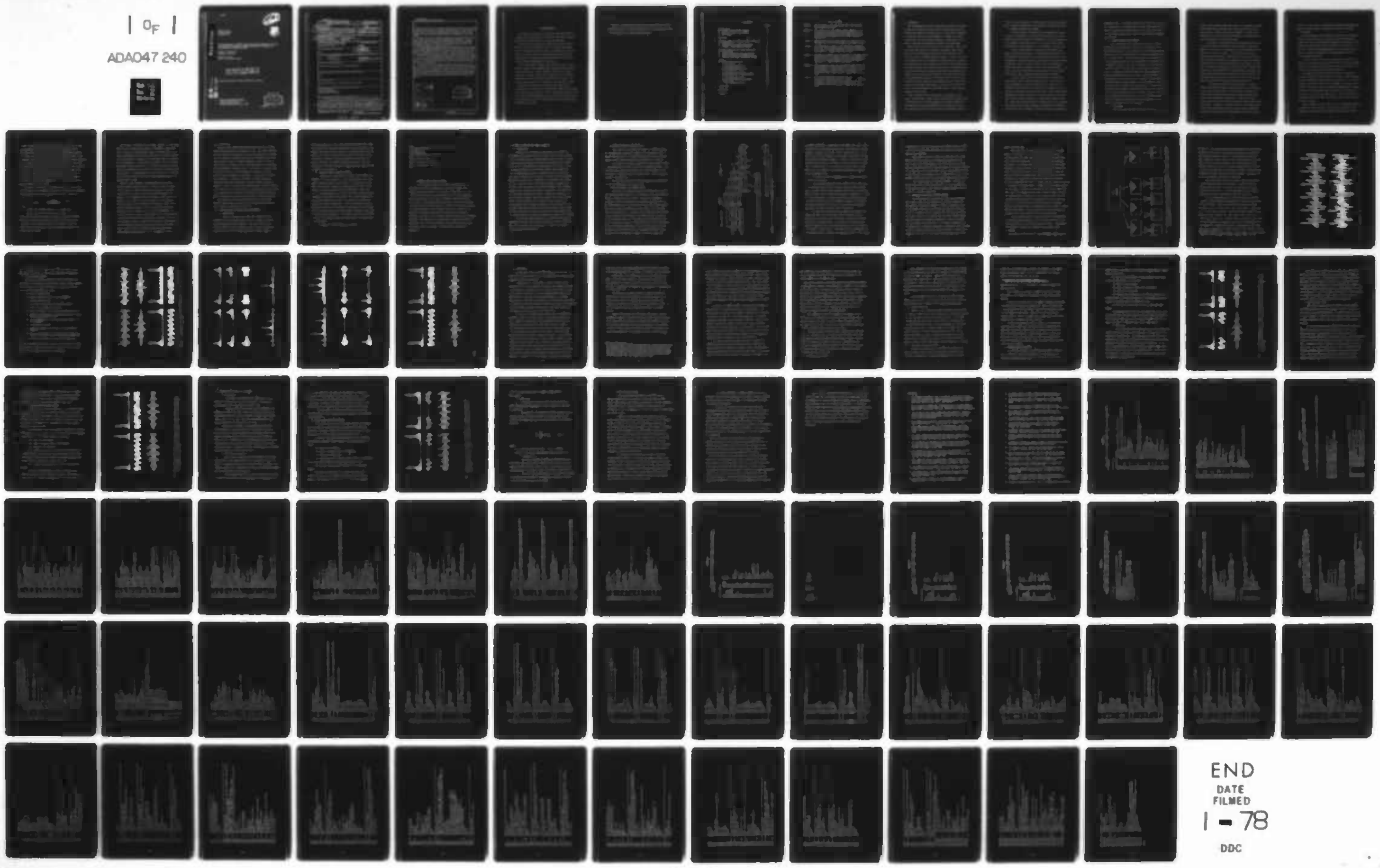
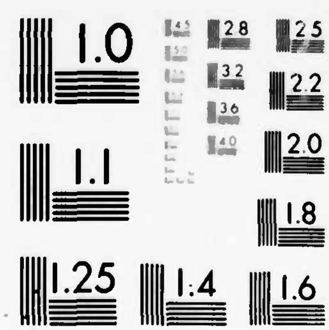
AD-A047 240 ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y F/G 17/2
THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS --ETC(U)
OCT 77 R J NIEDERJOHN, R A CURTIS
UNCLASSIFIED RADC-TR-77-310 NL

1 OF 1
ADA047 240



END
DATE
FILMED
1 - 78
DDC

A047



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963

RADC-TR-77-310
In-house Report
October 1977

THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS USE FOR THE STUDY AND DEVELOPMENT OF PROCESSING METHODS FOR ENHANCING THE INTELLIGIBILITY OF SPEECH IN NOISE

Russell J. Niederjohn
Marquette University

Robert A. Curtis
Rome Air Development Center

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER RADC-TR-77-310	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS USE FOR THE STUDY AND DEVELOPMENT OF PROCESSING METHODS FOR ENHANCING THE INTELLIGIBILITY OF SPEECH IN NOISE.		5. TYPE OF REPORT & PERIOD COVERED In-house Interim Report. 6 Jun [redacted] - 26 Aug 77, 6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Russell J. Niederjohn, Marquette University, Milwaukee, Wisconsin 53233 Robert A. Curtis (RADC/IRAD), Griffiss AFB NY 13441		8. CONTRACT OR GRANT NUMBER(s) N/A
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rome Air Development Center (IRAD) Griffiss AFB NY 13441		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 31011G 70550707
11. CONTROLLING OFFICE NAME AND ADDRESS Same		12. REPORT DATE Oct~~ober~~ 1977 13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same (12) 95p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Captain Robert A. Curtis (IRAD)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech Enhancement
Speech Intelligibility
Computer Speech Processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The enhancement of speech intelligibility in the presence of high noise levels is an important consideration in many practical communication situations. The work documented in this report describes the results of an examination of ~~several~~ four methods for processing speech so as to enhance its intelligibility in the presence of wideband random noise. Four ~~particular processing~~ methods were considered: ~~(a)~~ (1) INTEL, a method which involves processing in both the first and second order spectral domains; (b) spectral subtraction, ~~a method~~ which involves

(over)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

309 050 LB

20. (cont fr p i)

a simple subtraction of the average noise spectrum from the first-order spectrum; (3) minimum mean square error filtering, ~~a method~~ which involves filtering speech in such a way as to minimize the mean square error between a signal and its expected value in noise; and (4) methods based upon suppressing the frequency content of a speech plus noise signal between pitch harmonics of the speech signal.

To carry out a study of methods to enhance speech intelligibility in noise, two general-purpose computer processing systems were implemented. The first, ~~was~~ a terminal interactive system for ~~the~~ generation, analysis, and graphic display of synthetic voiced speech sounds, Through the use of this system, a provided considerable insight into the effect of various processing algorithms upon speech and upon speech in noise. has been effected.

The second computer processing system has been developed for the processing of real speech. It ~~This system~~ involves the use of a DDP-116 data converter and a Honeywell 6000 Computer. Communication between these two computers is by means of seven track magnetic tape. In use, this system facilitates the input, process, and playback of real speech utterances. Through this system, the effect of numerous processing algorithms upon normal speech in noise has been studied.

While both of these processing systems have been developed for, and applied to, a study of processing techniques for enhancing the intelligibility of speech in noise, the computer programs generated have purposely been made general purpose so as to facilitate their future use at RADC for other speech processing and signal processing tasks.

The effect of several processing algorithms (based upon the four methods mentioned) has been studied for numerous synthetic voiced speech sounds and for two 12-second real speech utterances. These two speech utterances were generated by a male talker: one utterance in a signal-to-noise ratio of +6 dB, the other utterance in a signal-to-noise ratio of -6 dB. Overall results indicate that while the greatest speech enhancement success has been achieved with the INTEL and the minimum mean square error filtering methods, the four methods studied each offer a significant potential for speech intelligibility enhancement in noise.

ii

ACKNOWLEDGEMENTS

The authors of this report are grateful to many persons who contributed to make this Summer Faculty Research Participation Program at the Rome Air Development Center a very valuable experience.

The help of Dr. Bruno Beek, Mr. Richard Vonusa and Mr. Michael Heffron of the Speech Processing Group at the Rome Air Development Center is sincerely appreciated. Also, the assistance of Mr. Albert Proctor, Mr. David Clark and Mr. William Doane who helped with the preparation of the computer programs and the other work associated with the Honeywell 6000 Computer is appreciated.

To not acknowledge the administrative assistance of Mr. J. Fred O'Brien, Director of the "USAF/ASEE Summer Faculty Research Participation Program" would be a major oversight. The work of Mr. O'Brien resulted in the smooth operation of the program and, in a major way, contributed to the technical success achieved by the project. The authors of this report are very thankful for the help of Mr. O'Brien.

The authors of this report would also like to express their appreciation for the interest and help of several other persons at RADC who contributed to this project. Three of these persons include Mr. John Huss, who coordinated the program at the local level; Mrs. Emma Kimpton who typed this report and Mr. Kirk Starczewski who prepared the figures contained in this report.

Finally, the authors wish to thank the United States Air Force and the American Society for Engineering Education who sponsored and coordinated the activities of the summer research program.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The understanding of speech contaminated by the presence of noise is an important consideration in many practical communication situations. Consider the airplane or helicopter pilot in a noisy cockpit attempting to communicate with ground based personnel; or consider the worker in the environment of noisy machinery attempting to communicate via telephone or other means with personnel outside the noise environment.

There are two cases for the speech in noise situation. First, there is the case where the noise is present at the speaker (1-6); second, there is the case where the noise is present at the listener (7-19). Each case presents a somewhat different situation. In the case of noise at the speaker, there is an opportunity to suppress the noise (relative to the speech) prior to its reception by the listener, thus resulting in an enhanced signal-to-noise ratio and hopefully more intelligible speech. In the case of noise at the listener, while there is no opportunity to suppress the noise, there is an opportunity to process the speech prior to its encounter with the noise. Conversely, in the first case, since the noise is already present with the speech there is no opportunity to process the uncorrupted speech signal. In the second case, since the noise is in the environment of the listener, there is no opportunity to suppress the noise level.

There are several types of noise which may contaminate a speech signal. These include: impulse noise, large amplitude sine-wave or sum of sine-wave noise, a conflicting speaker, and wideband random noise. The enhancement of speech intelligibility in the presence of each of these noise types is under study by the Rome Air Development Center (1-

3, 20). It is the purpose of the present study to consider the enhancement of speech intelligibility in the presence of wideband random noise.

A. SPEECH IN NOISE AT THE LISTENER

One obvious method for improving the intelligibility of speech in noise at the listener is to simply decrease the level of the noise. Although there exist methods for decreasing the noise levels produced by noisy equipment, these methods often do not reduce the noise levels sufficiently or are too expensive or too inconvenient to be practical.

Another obvious method for improving the intelligibility of speech in noise at the listener is to simply increase the power of the speech signal, thus resulting in a more favorable signal-to-noise ratio. Although such a technique may work in certain low-level noise situations; in high noise levels, the need to conform within a pre-established maximum sound level may prohibit the use of this simple technique. As a result, a method for enhancing the intelligibility of speech in high noise levels, without increasing the signal power, is desirable.

Several studies related to the intelligibility of speech in noise (at the listener) have been reported (7-19). In these studies, various techniques for processing speech (prior to reception by the listener) have been evaluated. Three particular techniques (8, 10, 11) have been recently reported and have been shown to offer intelligibility enhancement for speech in white noise at various signal-to-noise ratios and in one case in the environment of recorded power generating noise (9). These three techniques involve processing speech by: (a) high-pass filtering followed by infinite-amplitude clipping (8), (b) high-pass filtering (11), and (c) high-pass filtering followed by rapid amplitude

compression (8-9). At a signal-to-noise ratio of 0dB (noise at 90 dB (re. .0002 dy/cm^2)) an intelligibility of greater than twice normal speech is obtained by processing speech by any of these three methods. At other signal-to-noise ratios a similar intelligibility enhancement is also achieved.

B. SPEECH IN NOISE AT THE SPEAKER

One technique which may sometimes be successfully applied for enhancing the intelligibility of speech in the environment of noise at the speaker makes use of a noise-canceling or close-speaking microphone. In some cases such a microphone may not be sufficient because the noise is at too high a level. In other cases it may not be possible to use such a microphone because it's too inconvenient for the speaker, because the speaker is unable to wear such a microphone, or because the speech is already added to the noise when it is available for listening. As a result, a processing technique for enhancing the intelligibility of speech contaminated with noise is desired. Several techniques have been investigated for the case of noise at the speaker. These techniques include high-pass filtering followed by infinite amplitude clipping (4), a technique known as "INTEL" which involves gating the second-order spectrum and the subsequent retransformation back into the time-domain (1-2), and methods based upon the use of linear prediction (5-6). While the techniques investigated for this case have displayed encouraging results, a significant enhancement in the intelligibility of speech in noise at the speaker has not been achieved.

C. THIS REPORT

While both cases for speech processing in noise have many

practical applications and while both cases are of considerable interest, greater application and interest exists within the Air Force for the later case (i.e., speech in noise at the speaker). As a result, the remainder of this report and the work performed in its preparation are directed specifically toward this later case.

The research work documented in the following sections of this report has included an analysis and study of methods for processing speech so as to result in an enhancement of its intelligibility in the presence of noise at the speaker. In this direction two specific approaches have been pursued. The first approach involves the development of interactive software for the analysis and study of processing techniques applied to synthetically generated voiced speech. For this study, interactive programs have been developed for the Honeywell 6000 accessed via a Tektronix 4002A CRT for the generation, processing, and graphic display of synthetically generated voiced speech sounds. This interactive system is described in Section III of this report and the results obtained with this system are described in Section V.

The second approach involves the development of a general purpose, batch-oriented, speech processing system for real speech and its use in the study of speech in noise processing techniques. This system involves magnetic-tape communication between the DDP-116 data converter and the Honeywell 6000 computer. Programs for analog to magnetic-tape conversion, previously developed by Captain Robert Curtis, are utilized. Programs for the input and output of this magnetic-tape to/from the Honeywell 6000 computer and general purpose programs for processing the data contained on these magnetic-tapes are described in

Section IV of this report. In Section V the results obtained using this system for processing speech contaminated with noise are described.

II. DISCUSSION

The most successful present technique for enhancing the intelligibility of speech in noise at the listener is the technique known as "INTEL" (an acronym for INTelligibility Enhancement by Liftering). This technique has been developed under the direction of the RADC (1-2). While only a small speech intelligibility gain has been achieved by this method, a significant enhancement of the listenability of speech in noise has been demonstrated.

It is one purpose of this work to investigate the operation of the INTEL technique through both an examination of the results of processing synthetic voiced sounds and through the processing of real speech. The results of this examination are presented in Section V.

One problem with the INTEL technique is that it requires four Fourier Transformations, two forward and two reverse. Since calculating a Fourier Transform is computationally time consuming, it is of interest to explore methods which require fewer transforms. As a result, three additional approaches to speech intelligibility enhancement are explored in this work with preliminary results documented in later sections of this report. These three approaches are: (a) spectral subtraction, (b) minimum mean square error (MSE) filtering (21), and (c) methods based upon pitch tracking.

Spectral subtraction is of interest since it is computationally and conceptually relatively simple. It involves the subtraction of the estimated noise spectrum from the transform of the speech plus noise

signal. After subtraction, the resulting spectrum is retransformed into the time-domain. Support for this technique can be gained from an intuitive analysis of speech plus noise and from its seeming similarity to "gating" which is performed on the second-order spectrum in the INTEL technique. One problem with subtraction is that it requires some determination of both the magnitude and the spectrum of the noise. In the INTEL technique, the magnitude of the noise is automatically accounted for (when gating to zero is used) and the spectrum of the noise is assumed flat. In the results presented in Section V for spectral subtraction it is assumed (as a first approximation) that the noise has a flat frequency spectrum (i.e., white) with a magnitude equal to the average first-order spectrum magnitude above 2.5 KHz.

A method for determining a filter which minimizes the mean square error between a signal embedded in noise and its estimated value, assuming the signal and noise are uncorrelated, can be shown to be given by (21):

$$H(jw) = \frac{Sss(w)}{Sss(w)+Snn(w)} \qquad \text{(eq. 1)}$$

where:

Sss(w) = the estimated spectrum of the signal, and

Snn(w) = the estimated spectrum of the noise.

Using the longtime average speech spectrum for Sss(w) and the measured spectrum of the noise for Snn(w), the transfer function H(jw) can be calculated. Then, multiplying H(jw) by the spectrum of the input speech signal and performing an inverse transformation results in the processed speech output.

This method is attractive for two reasons. First, it is an optimum method (in the least mean square sense) for separating a signal from noise. Second, it is computationally simple to implement since it is a simple filtering process which can be performed in the frequency domain after transformation, or perhaps on the time-domain signal using digital filtering techniques. A complication of the method is that it requires an estimate of the noise spectrum. It is unclear at this point how the results of this method vary as a function of the error in estimating the noise spectrum. In the experiments described in Section V it is assumed that the noise spectrum is flat with an average magnitude equal to the magnitude above 2.5 KHz.

Speech in noise enhancement through methods based upon pitch extraction seem intuitively attractive. This is a result of the fact that the energy of a speech signal exists at harmonics of the pitch frequency while the energy of noise is distributed throughout the spectrum. From a knowledge of the pitch frequency, those lines in the spectrum not at harmonics of the pitch frequency can be suppressed leaving, hopefully, speech enhanced in the presence of noise. Several problems influence this method. There is the problem of accurately tracking the pitch frequency, even in a non-noise environment. There is, on the one hand, a desire to analyze speech over a long time segment to gain as much information about the signal as possible. There is, on the other hand, a desire to analyze a short interval so that changes in the pitch frequency during the analysis interval will not be significant. A recent report by Parsons and Weiss (3) suggests that the optimum segment size should be 40.8 msec. A segment of 51.2 msec is

used in this work.

Previous results obtained with methods based upon pitch tracking have not been particularly encouraging (1-2, 20). It has been suggested that even when the pitch frequency is accurately determined, for example from the original uncorrupted speech signal, such methods have not been shown capable of significant speech enhancement in noise. The use of comb filtering, for example, has not been found effective for improving the intelligibility of speech in noise. A discussion of the use of comb filtering for speech in noise enhancement is contained in a recent report by Weiss, Aschkenasy, and Parsons (1). Another report, by Weiss and Aschkenasy (2), discusses an experiment with pitch tracking in which good results were reported when the pitch frequency was found adequate. Details of this method are somewhat sketchy. However, pointed out in this report are the difficulties in determining pitch, particularly at low signal-to-noise ratios, and the distortion produced by inaccurate pitch tracking and by analyzing unvoiced speech by a harmonic analysis.

An interesting method for pitch tracking and some interesting results obtained, from a relatively crude processing method based upon pitch tracking, are described in Section V.

III. COMPUTER SYSTEM FOR SYNTHETIC SPEECH PROCESSING

A. INTRODUCTION

In order to provide some insight into the speech in noise situation and into processing speech in noise, a terminal interactive speech processing system was developed. This system makes use of a Tektronix 4002A CRT character/graphics computer terminal connected via telephone modem with the Honeywell 6000 Computer System at RADC. A

program developed by Mr. David Clark of RADC which provides FFT, IFFT, data input/output, and graphic capability was used as a starting point. Added to this program is the capability for analyzing speech by the four methods (with several variations each) discussed in the last section. In addition, a program for generating synthetic voiced speech in a format acceptable to the analysis program was developed. These two programs are described in the next two subsections with program listings in Appendix A and Appendix B.

B. SYNTHETIC SPEECH GENERATION PROGRAM

A program for the generation of synthetic voiced speech called "SPEECH," is listed in Appendix A. This program assumes a sum of three damped sine-waves model for voiced speech. It requests the following input: (a) amplitude, frequency (Hz), and damping rate (Hz) for three formants; (b) pitch period (msec); and (c) RMS signal to RMS noise ratio. The program generates 51.2 msec of the speech signal (non-pitch synchronous, with sample values spaced at 100 us), adds random noise of the specified signal-to-noise ratio, and provides the capability for viewing any generated sample values. Finally, the program outputs the generated speech plus noise samples into a (previously defined) random file, with name specified in the RUN statement (program line number 0010 in program listing, Appendix A). The format of this file is acceptable as input to the synthetic speech processing program. The program runs under GCOS FORT.

A sample run of this program is as follows:

```
*RUN
INPUT: AMPL,FREQ,ALPHA, FOR ALL 3 FORMANTS
=1.,730.,27.
=.5,1090.,28.5
=.04,2240.,46.5
INPUT PITCH PERIOD IN MSEC
=9.
NOISE?(1-YES, 0-NO)
=1
INPUT S(RMS)/N(RMS) IN DB
=0.
VIEW ANY SAMPLES?(1-YES, 0-NO)
=0
```

C. SYNTHETIC SPEECH ANALYSIS PROGRAM.

A very flexible program for the analysis and display of general time-waveforms has been developed by Mr. David Clark of RADC. This program provides the capability for the input and output of data files, for FFT and IFFT analysis, for the display of data, for data generation under cursor control, and for several mathematical operations between two data files. This program has been modified to include the processing of speech by several methods based upon (a) INTEL, (b) spectral subtraction, (c) minimum mean square error filtering, and (d) pitch extraction. In addition, the updated program permits the capability for examining (plotting) the results of any intermediate subprocess within each processing algorithm. The program runs under GCOS TFORT, is named M2 (following an appellation M1 for the previous program), and contains instructions upon initiating a run of the program. A listing of M2 is contained in Appendix B.

IV. COMPUTER SYSTEM FOR REAL SPEECH PROCESSING

A. INTRODUCTION

While the use of synthetic voiced speech is very useful for providing an understanding of both the general speech in noise situation as well as specific speech processing methods; the real proof of a speech in noise enhancement technique is how well it performs for real speech. In an effort to study speech processing methods on real speech a set of programs were developed for the Honeywell 6000. These programs run under the GCOS operating system to facilitate the input of data, the processing of that data, and the output of results. Since the Honeywell 6000 does not provide A/D or D/A capability, the DDP-116 data converter is used. Using the DDP-116, input speech waveforms are sampled (A/D converter) and these samples are written onto magnetic tape. Having a magnetic tape of samples to be output to the DDP-116, a time signal can be constructed using the D/A converter. Programs for the DDP-116 for A/D conversion to magnetic tape and for magnetic tape to D/A conversion, written by Captain Robert Curtis of RADC, were used. Making use of these magnetic tapes of speech data, programs have been written for the Honeywell 6000 Computer System to read-in these magnetic tapes, process the data they contain, and output the results on a second magnetic tape in a format suitable for input (and subsequent D/A output) by the DDP-116.

The software system developed for the Honeywell 6000 has been purposely written in a very flexible, very general, and well commented way so as to facilitate its future use at the RADC for any general speech or signal processing task.

B. SPEECH INPUT/OUTPUT WITH THE DDP-116

Input signals to the DDP-116 are sampled at a 10 KHz rate (10-bit sign-magnitude) and written to magnetic tape in records of 1024 samples (102.4 ms/record). The DDP-116 generates three tape words per sample in a tape format established by the hardware of the system.

For output, the software of the DDP-116 requires magnetic tapes of 1024 samples/record written in the proper sign-magnitude format. The output software created for the DDP-116 can read these magnetic tapes and generate a time-waveform from the sample values. The D/A converter is 10-bit, sign-magnitude.

For both input and output the DDP-116 requires a one word zero record at the beginning of any magnetic tape.

C. DDP-116/HONEYWELL 6000 COMMUNICATIONS

The media for communication between the DDP-116 data converter and the Honeywell 6000 computer is 7-track magnetic tape. The Honeywell 6000 has a memory word size of 36-bits, utilizes two's complement arithmetic, and when communicating with a magnetic tape unit utilizes six tape-words/CPU word. The DDP-116 has a 16-bit word, utilizes sign/magnitude arithmetic, and communicates with a magnetic tape unit with three tape-words/CPU word. A method for tape to data-word and data-word to tape translation was developed for the Honeywell 6000. The essence of the processing necessary for this magnetic tape communication situation is illustrated in Figure 1. Each successive word read from magnetic tape by the Honeywell 6000 consists of two successive words written by the DDP-116 (6 tape-words/CPU word for the Honeywell 6000 and 3 tape-words/CPU word for the DDP-116). Figure 1 illustrates (right) the

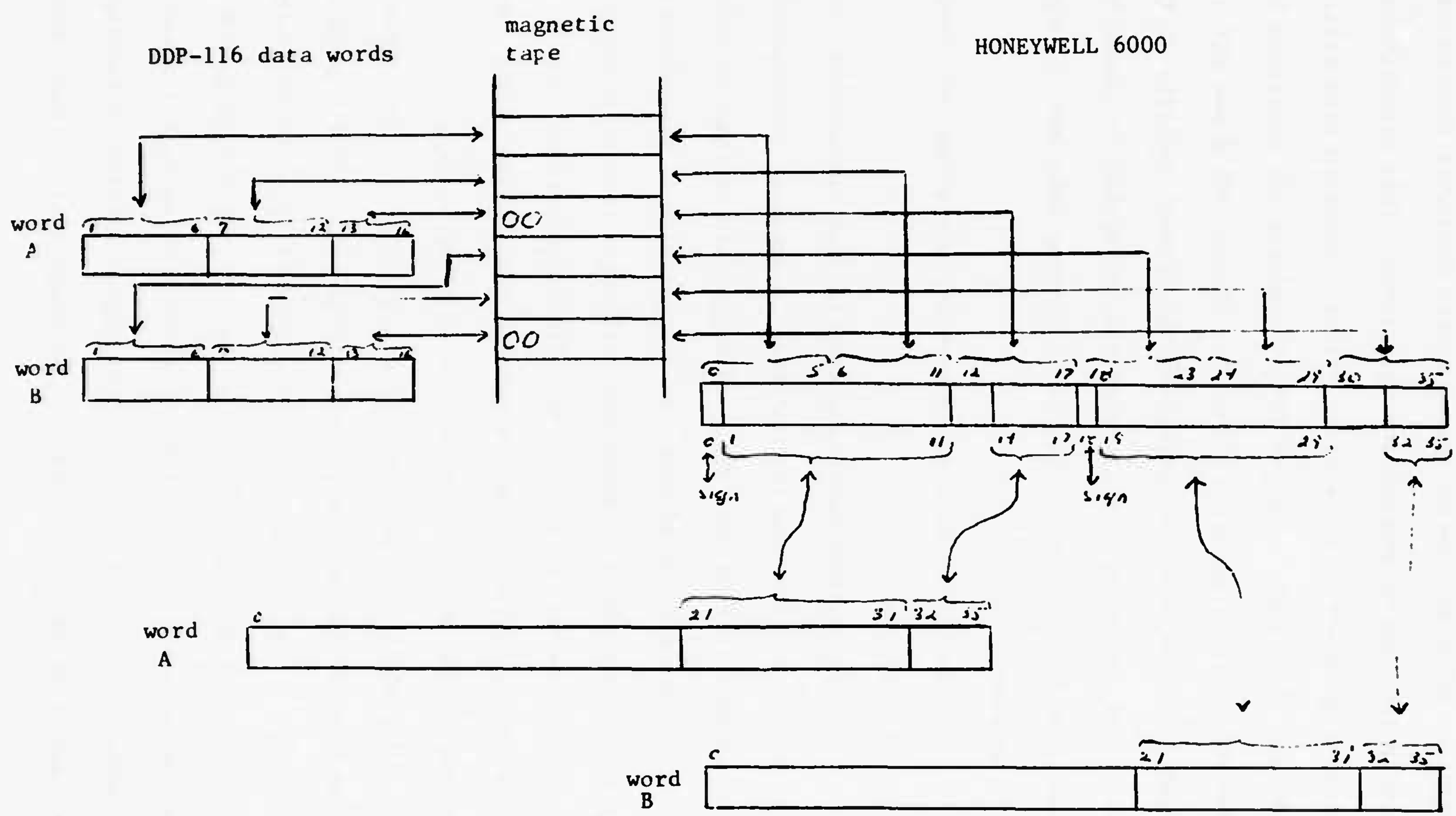


FIGURE 1. An illustration of the DDP-116/Honeywell 6000 magnetic-tape communication method,

translation which must be performed by the Honeywell 6000 to map a tape-read word into two data-words and vice-versa. This translation process requires a sequence of two events. First, tape-word bits 1-11, 14-17, 19-29, and 32-35 must be mapped into the proper bit positions in the two data-words as illustrated in Figure 1. Second, bit 0 and bit 18 of the tape-word must be tested to ascertain the correct polarity of the data-word. If the data-word is to be positive, no action is taken; if the data-word is to be negative, the corresponding data-word is negated (two's complement sense).

To translate a pair of data-words into a tape-word requires the reverse of this process.

The subroutines which are used to read and write a tape record by the Honeywell 6000 have been written in assembly language (GMAP). These programs (called RTB, WTB, and WTBZ) were written in consultation with Mr. Albert Proctor of RADC and are included in Appendices C, D, and E. The processes used to translate data-words to tape-words and tape-words to data-words are written as FORTRAN subroutines (called DATTAP and TAPDAT respectively) and are part of the program called PROCESS described in the next section and listed in Appendix H.

One additional problem exists with magnetic tape communication between the DDP-116 and the Honeywell 6000. The DDP-116 outputs a parity word at the end of each record which is inconsistent with that expected by the Honeywell 6000. As a result, when reading DDP-116 generated magnetic tapes on the Honeywell 6000, a parity error signal to the operator's console results. This signal requires acknowlegement by the computer operator in order for the program to proceed. When reading

a large number of records by this process, the operator tends to prefer to abort the job rather than acknowledge every parity error (as many as 2,000 during a long run).

There exists a software system on the Honeywell 6000, called UTILITY, which may be used to read magnetic tapes ignoring parity errors. Using this system a magnetic tape can be copied onto another magnetic tape (with proper parity for reading) or can be written directly into a data file. In the present work a copy to a second magnetic tape is utilized. This second magnetic tape can then be left as part of the Honeywell 6000 system and the program "UTILITY" need not be used again.

The program UTILITY runs under GCOS CARDIN. A listing of a batchjob for a tape-to-tape copy using utility is shown in Appendix F. A second batch program to read a "UTILITY" generated magnetic tape into a data file is shown in Appendix G.

D. HONEYWELL 6000 SPEECH PROCESSING PROGRAM

The main program for processing real speech on the Honeywell 6000 computer system is called PROCESS and is listed in Appendix H. This program runs under GCOS CARDIN. It reads-in speech data from a file (defined in the batchjob), processes that data, and outputs the results to a magnetic tape (also defined in the batchjob) in a format acceptable for input by the DDP-116. The processing program assumes that some method of overlapping time-windows (of 1024 data samples) is to be applied to the input data. In the present implementation of the program only a triangular time-window is implemented. This processing program is purposely written very flexibly to permit its future use for speech in noise work, for other speech processing work, and for other

signal processing work.

The program has four hierarchal levels of subroutine program depth as illustrated in Figure 2. At the highest level, the main program performs data input, data output, overlapping time windowing, output amplitude normalization, and calls to the various processing techniques. At the second level, the program includes the processing techniques. At the third level, the program includes the subprocessing steps. The fourth level (not illustrated in Figure 2) includes subroutines called by the subprocessing steps. The particular processing techniques to be called may be inserted with appropriate subroutine calls at program lines 310 to 436. A program run may include the performance of any number of processes (defined by NPROC, line 102, maximum value of 20), any number of records to be processed by each process (defined by NREC, line 105), starting with any record (defined by NFREC, line 107). The program creates a magnetic tape of the processed output with each successive processed speech output separated by ten zero records which result in a 1 sec silence gap for use when listening or recording. In addition, the program prints a listing showing each processing technique executed, the numbers of the first and last tape records created, and the CPU execution time required for each processing method.

In its present version, the program includes an implementation of eleven processing techniques. It also includes twenty-three third level subroutines for call by each processing technique. The program is well commented to permit ease in modification and future usage.

V. RESULTS

An analysis of several speech in noise enhancement methods has been

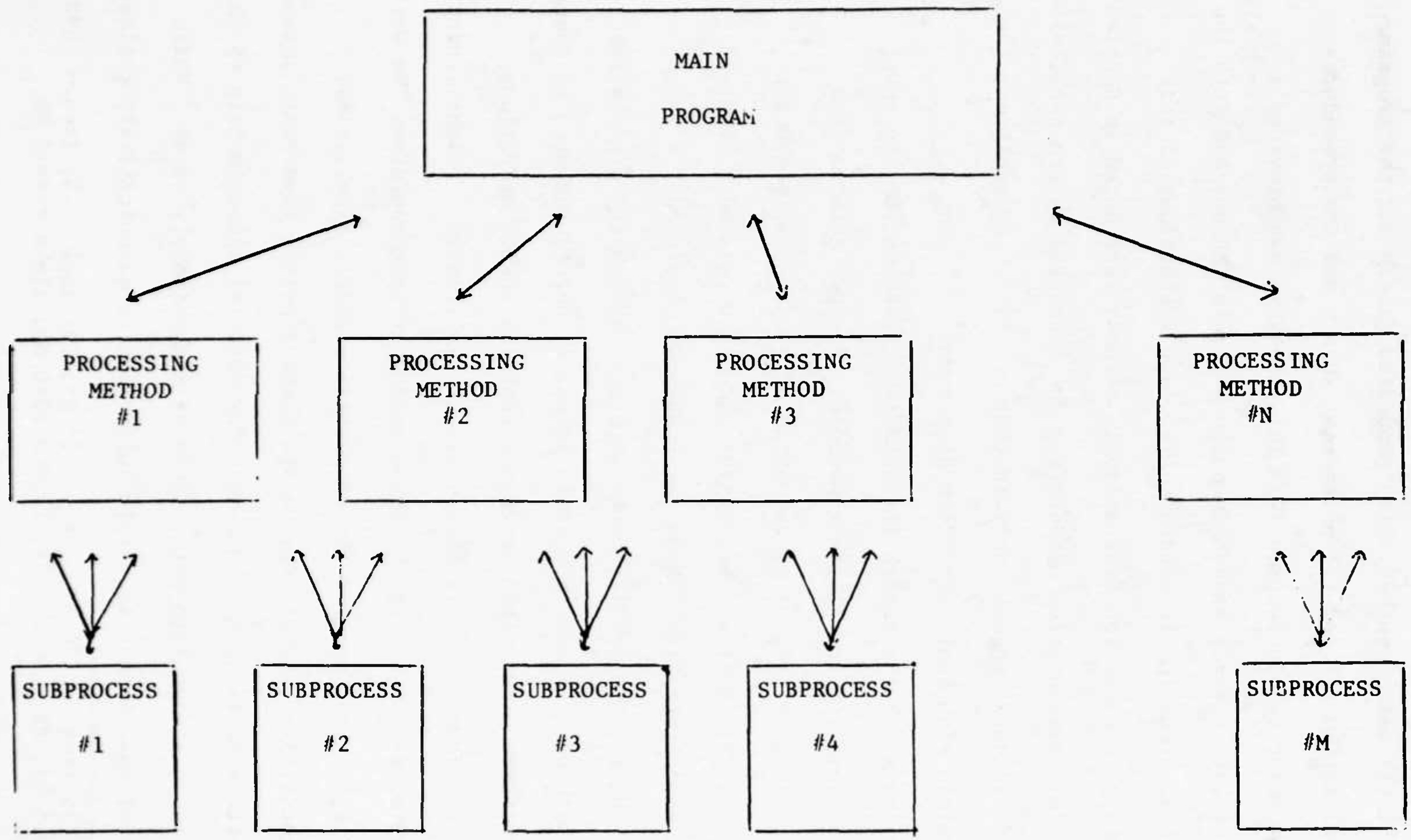


FIGURE 2. An overview of the real speech processing program "PROCESS." N is the number of processing techniques implemented by the system (presently twelve), M is the number of subprocessing steps implemented by the system (presently twenty-four).

carried out: (1) using synthetically generated speech and the programs discussed in Section III and (2) using real speech and the procedures and programs described in Section IV. This analysis has provided a basic insight into several speech in noise analysis methods and into the speech in noise situation in general. The four subsections of this section describe each of the four analysis methods introduced in Section II and contain representative output from the synthetic speech generation (SPEECH) and synthetic speech analysis (M2) programs. In addition, the results obtained with real speech are discussed.

For the synthetic speech illustrations, results for the vowel /a/ have been selected for all four methods. Formant data is from Peterson and Barney (22)*. A pitch period of 9 ms was selected and signal-to-noise ratios of ∞ (no noise) and 0dB were used. Figure 3 illustrates the output from SPEECH for these two conditions.

For the intelligibility tests with real speech two utterances are used. Both were spoken by Captain Robert A. Curtis and each is about 12 sec in duration. The first utterance contains speech as follows: "Testing...one...two...three...four...five...six...seven... eight...nine... ten... we were away a year ago." The second utterance contains, "We were away a year ago...testing...one...two...three...four...five...six... seven...eight...nine...ten... may we all learn a yellow lion roar...Hawaii." The first utterance is in a signal-to-noise ratio of approximately +6 dB; the second is in a signal-to-noise ratio of approximately -6 dB. While this corpus of speech data is hardly sufficient to quantitatively evaluate

*F1: ampl.= 1, freq. = 730 Hz, alpha = 27 Hz; F2: ampl.= .5, freq.= 1090 Hz, alpha = 28.5 Hz; F3: ampl.= .04, freq.= 2240 Hz, alpha = 46.5 Hz

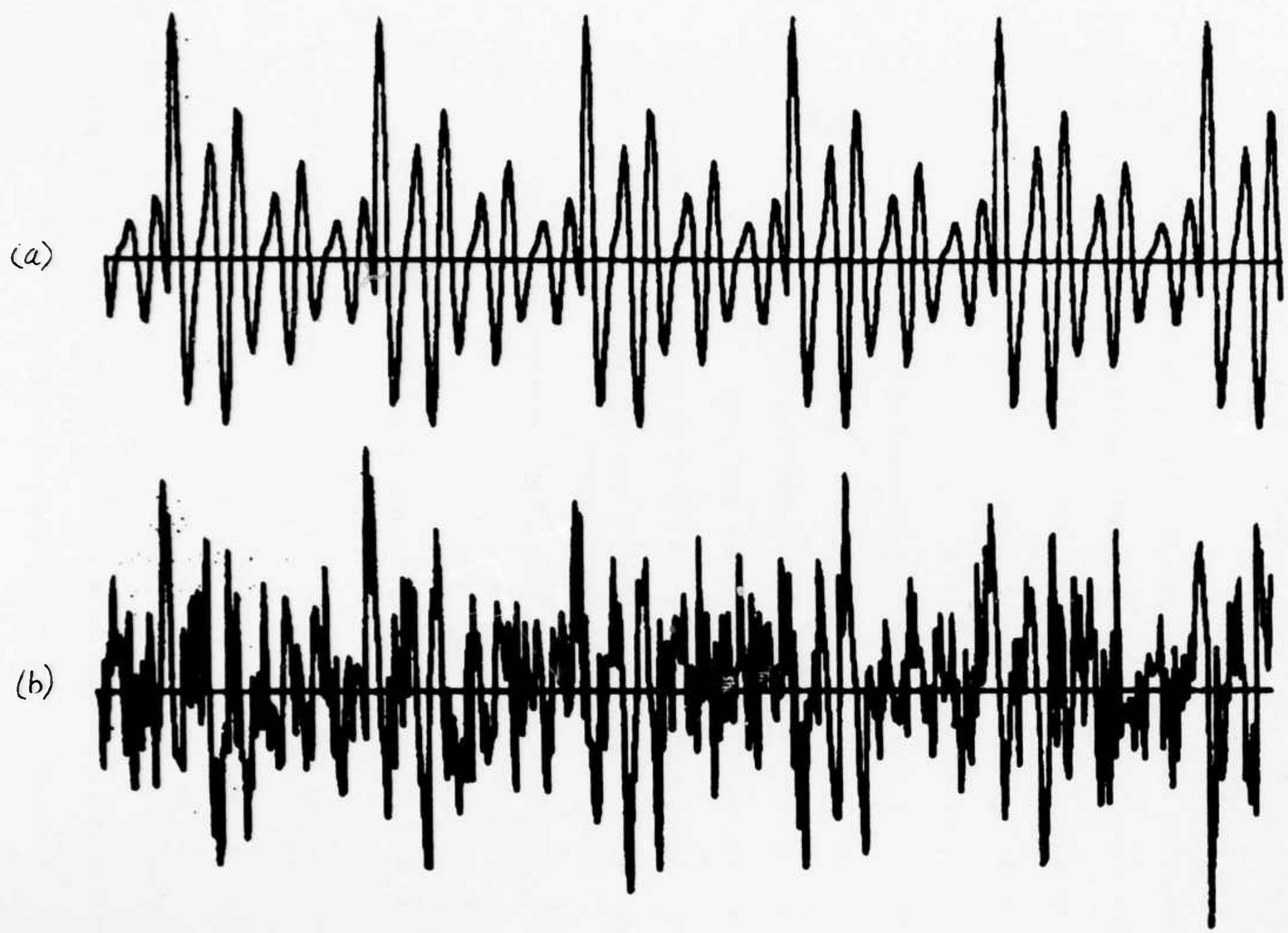


FIGURE 3. Time waveform of the synthetic vowel /a/ as generated by "SPEECH"; (a) no noise, (b) signal-to-noise ratio of 0dB.

speech intelligibility, it is sufficient to make a preliminary study of the processing methods and to make several qualitative observations.

A. INTEL PROCESSING METHOD

The INTEL algorithm is described in two recent reports (1-2). This author's interpretation of the algorithm from these reports indicates that the analysis of a single time-frame of 512 points (51.2 ms) consists of the following sequence of steps:

(a) Input 512 time samples;

(b) Apply a triangular window to the time samples;

(c) Perform a 512 point FFT;

(d) Set to zero the magnitude spectrum above 2.5 KHz;

(e) Square root the magnitude spectrum;

(f) Reverse the signs of all odd numbered magnitude harmonics;

(g) Perform a 512 point FFT on the magnitude spectrum as a real signal with zero imaginary part;

(h) Set the magnitude of the five low frequency harmonics to zero (gating);

(i) Perform a 512 point IFFT;

(j) Reverse the signs of the real part of all odd numbered harmonics;

(k) Square the real part of the spectrum, make it a magnitude, and restore the phase from the original time waveform; and

(l) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the result after each processing step listed in the previous paragraph are illustrated in Figure 4 for: (1) no noise (left) and (2) a signal-to-noise ratio of OdB (right).

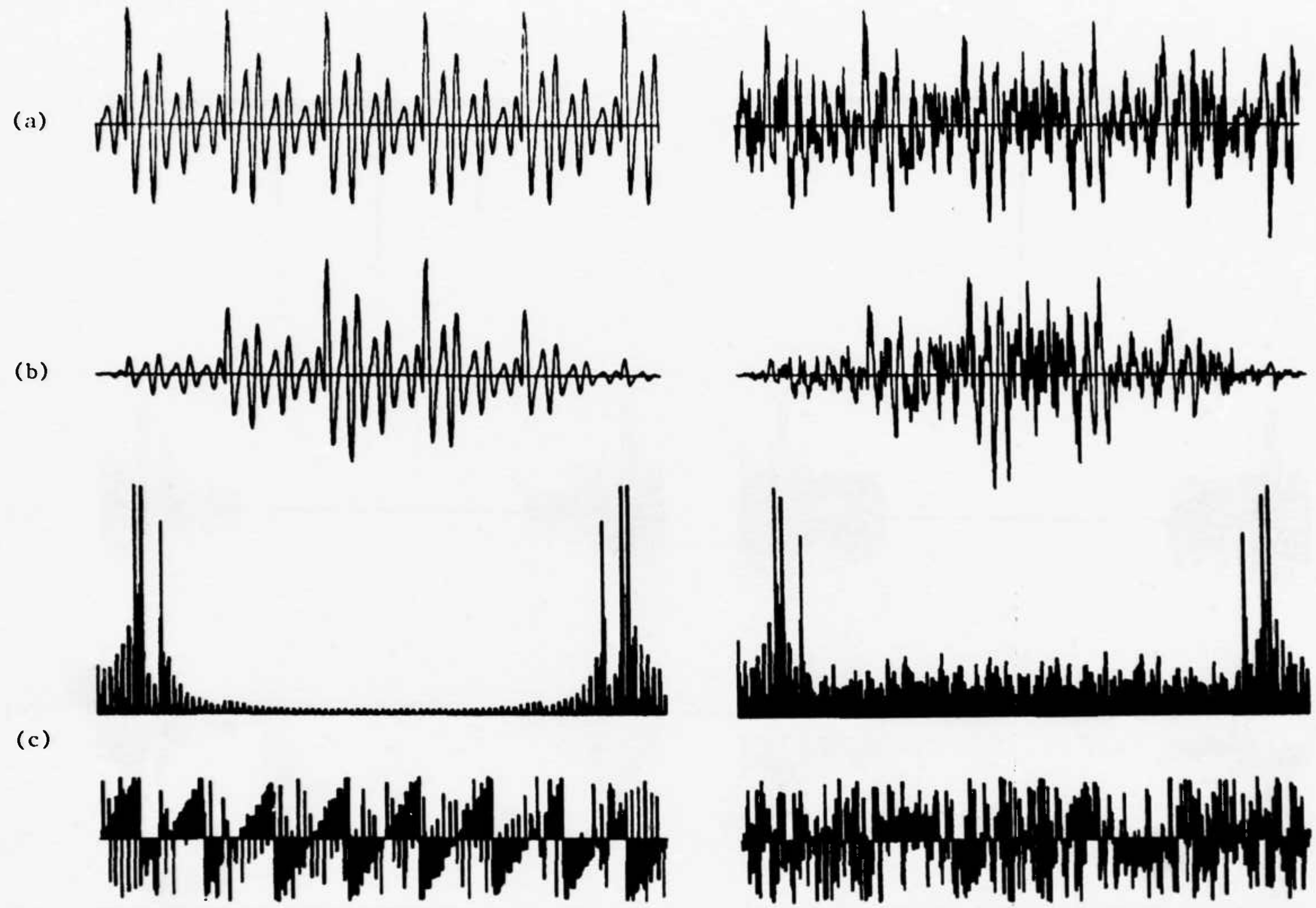


FIGURE 4. The Intel process applied to the synthetically generated vowel /a/. Each waveform shown is after the corresponding step listed in the first paragraph of Section V-A; left is for a S/N = ∞, right is for a S/N = 0dB.

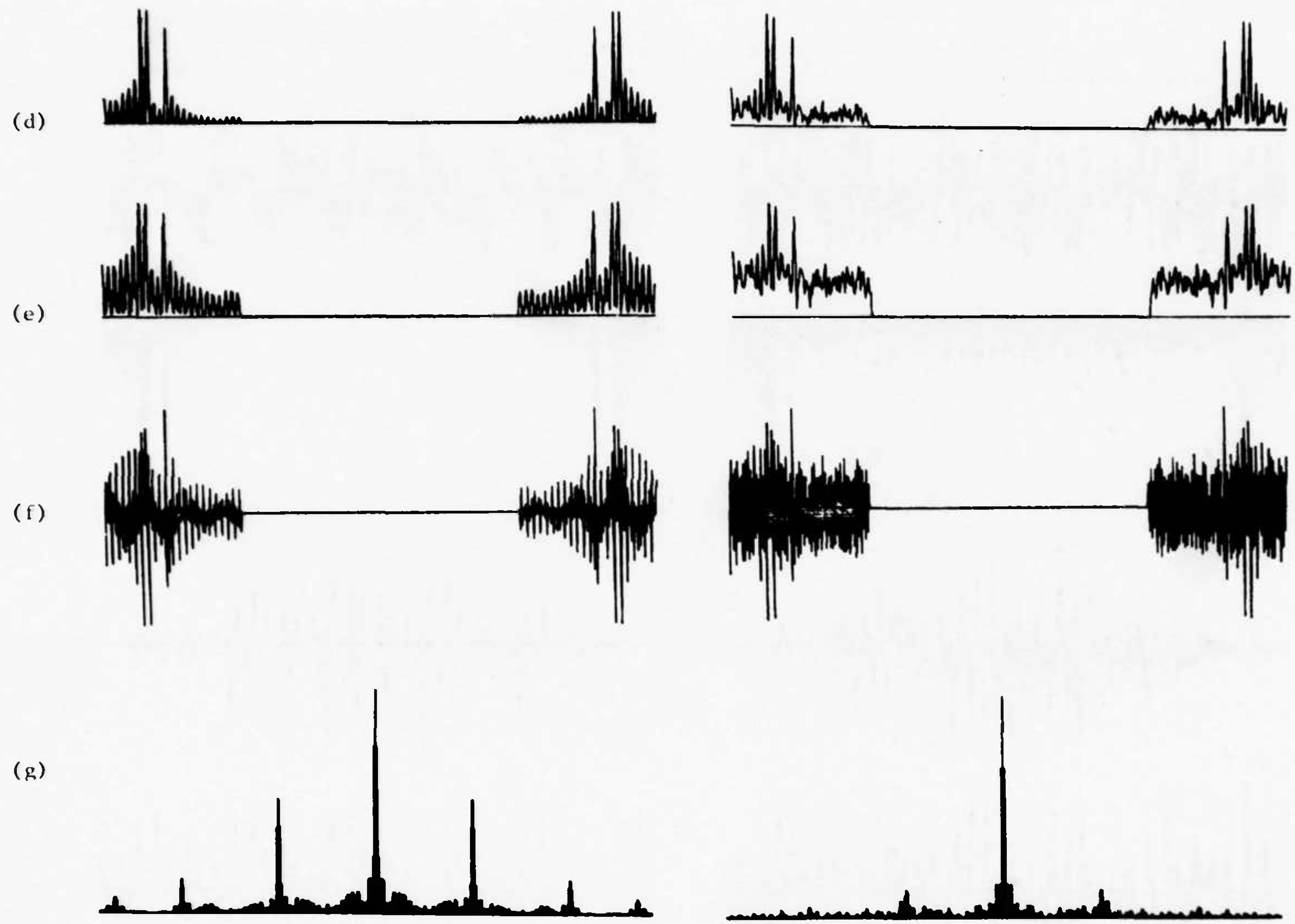


FIGURE 4. Continued

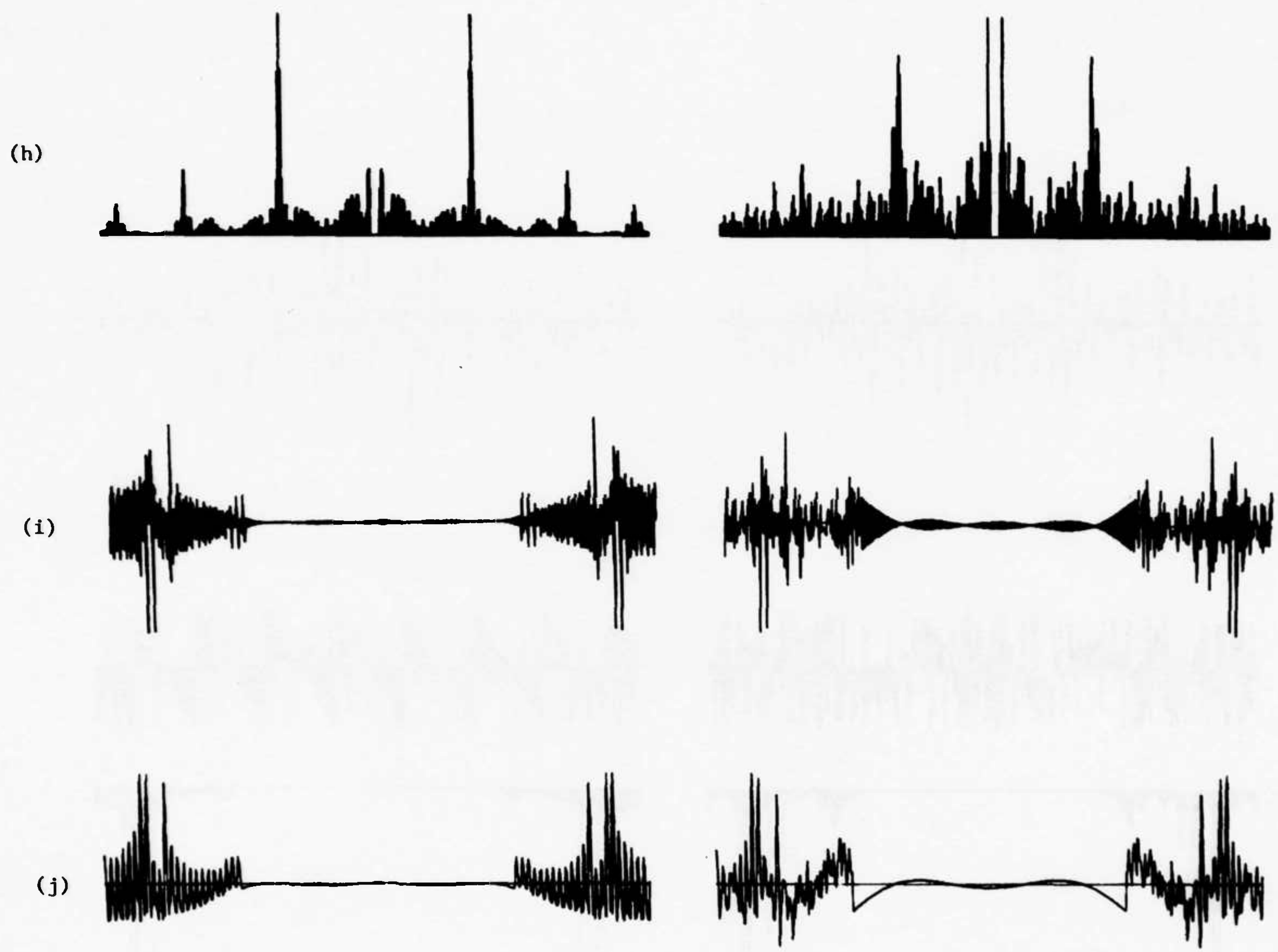


FIGURE 4. Continued

(k)

(l)

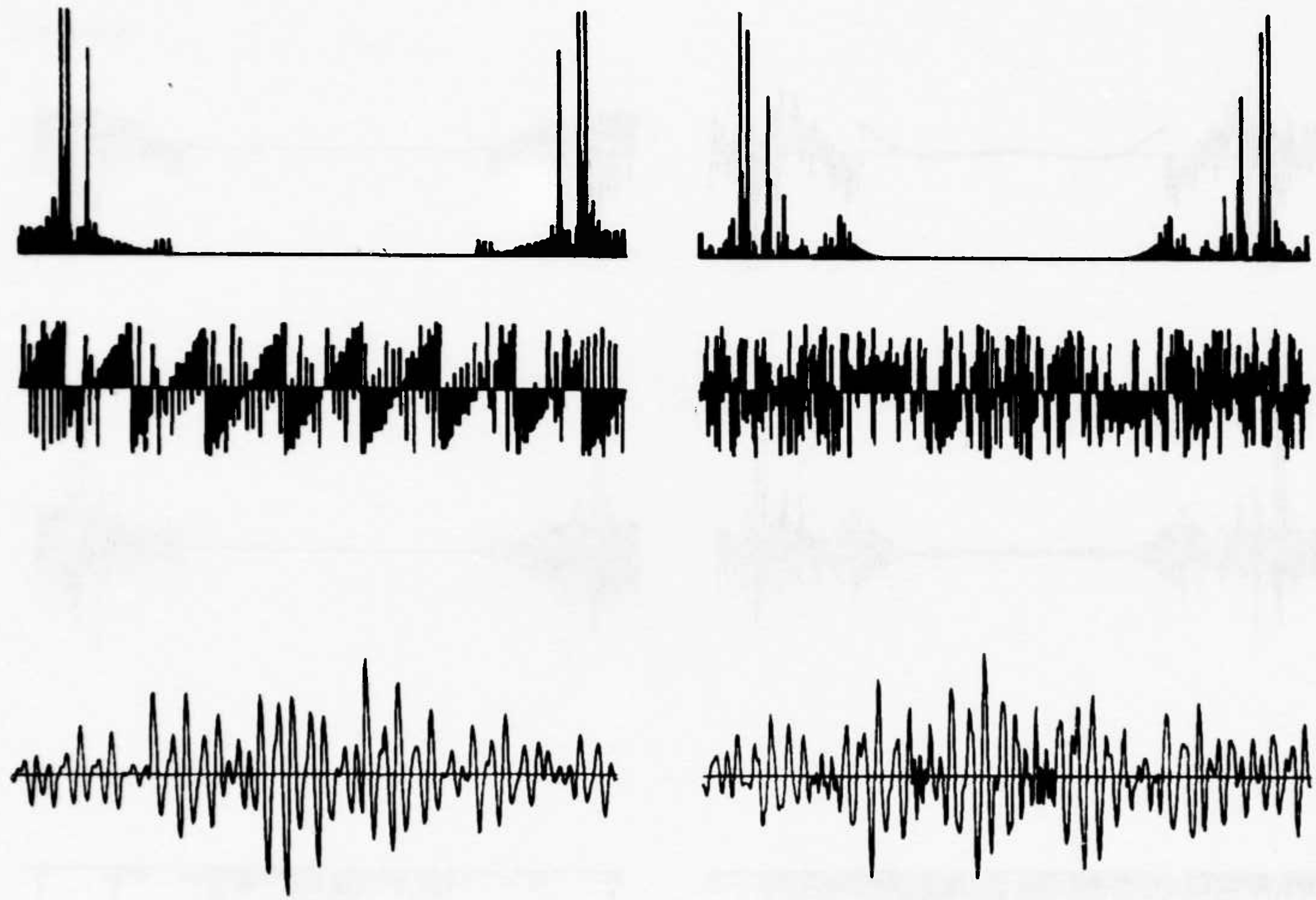


FIGURE 4. Continued

1. Discussion

A discussion of the overall INTEL speech processing method can be found in two recent reports (1-2). The discussion here assumes general familiarity with the INTEL method.

From Figures 4a - 4c the effect of noise on both the time waveform as well as on the magnitude and phase of the frequency spectrum can be observed. From Figure 4c, it is noticed that the effect of noise on the frequency spectrum is to add a randomized baseline shift to the magnitude of the spectrum and a seeming randomization of the phase characteristic. The phase of the spectrum of the speech plus noise signal is apparently not without information, however, since greater intelligibility has been achieved with INTEL by restoring the input phase rather than discarding it in producing the processed output (20).

In step e of the INTEL process a square root of the magnitude spectrum is taken. In a recent report (2), Weiss and Aschkenesy go to great lengths to demonstrate that a justification for this square root operation centers around the fact that such an operation causes an increased fraction of the noise energy (relative to the speech energy) to be concentrated in the region near the origin in the second order spectrum (Figure 4g) than would otherwise be present. It is therefore argued (2) that such a square root operation makes it possible to more completely remove the noise. What is unclear to the present author is whether the "extra noise" energy found near the origin in the second order spectrum is due to the original noise being moved or, more simply, the first order spectrum flattening which is caused by the square root operation. Following this latter argument, it is then not a desirable

feature of square rooting which causes increased energy near the origin of the second order spectrum, but rather a consequence of square rooting.

Steps f and j of the INTEL process both involve reversing the signs of all odd numbered harmonics. The only effect of this pair of sign reversals is to reverse the scale of the axis of the second-order spectrum. A computational advantage has been suggested as a justification in a recent report (2).

The essence of the INTEL technique is observed in Figure 4g which shows the second-order spectrum for the synthetic vowel /a/ in signal-to-noise ratios of ∞ and 0dB. Note the large peak at low "frequency"* in the second order spectrum for the signal plus noise. As shown in Figure 4h, this peak is removed by "gating" in the INTEL process.

Figure 4i illustrates the first-order spectrum after "gating" and Figure 4j after the sign reversals are "reversed". Comparing Figure 4j with Figure 4e (the spectrum of the signal before gating), the effect of the gating operation upon the spectrum is observed. In the absence of noise, the effect appears to be primarily a shift in the amplitude of all magnitude harmonics. In the presence of noise, the effect appears

*The use of the term "frequency" in describing the second-order spectrum may be confusing. While it is true that the units of the horizontal axis for the second-order spectrum are not Hz when carried from the original time waveform; the units are Hz if we consider the first-order spectral signal like a time waveform as seems to be a convenient way for viewing the INTEL Technique. In any case, a clear distinction of which spectrum is being discussed when using the term "frequency" will be made.

to be a removal of the baseline (Figure 4e) by a downward shift plus an added distortion in the frequency range above 2.5 KHz. A more careful observation of Figure 4j, however, reveals a decrease in spectral energy at about 1.5 KHz and a marked increase above about 2 KHz. This distortion was observed in a recent report (2) and can be theoretically shown to be a consequence of the gating operation in the second order spectrum. A method for automatically compensating for this distortion is described in a recent report (2). Time did not permit the addition of this compensation algorithm to the INTEL procedure programmed in the present work.

In step k of the INTEL procedure, the spectrum of the signal is squared prior to retransformation back into the time-domain. The justification for this step is an attempt to compensate for the square-root operation in step e(2). This author wonders why it is thought necessary to compensate for the square-root operation. There is little reason to suspect that attempts to preserve the original frequency spectrum are necessarily beneficial, or even desirable. The now classic work of Licklider and Pollack (27), as well as work of others, clearly shows that speech can withstand severe frequency as well as amplitude distortion without a significant loss of intelligibility.

To test the need for the square operation, a comparison of speech processed by the INTEL procedure, both with and without the square operation were examined by listening. Speech processed without squaring sounded very much like the unprocessed original. This indicates that apparently the square operation is essential to the INTEL process.

This surprising result (surprising to this author anyway) leaves open several unanswered questions regarding the use of the square root and square operations in the INTEL process.

One question remaining is why square rooting (and subsequent squaring) has been observed to improve the intelligibility of speech in noise relative to not square rooting. One argument is because the result of these two operations is to modify the output frequency spectrum in such a way as to emphasize lower frequency components relative to higher frequency components. Since lower frequency components tend to best survive the noise, speech square rooted (followed by squaring) would be expected to sound better (enhanced listenability) in noise. Whether this process results in enhanced intelligibility remains to be demonstrated.

The argument of the previous paragraph would suggest that the root used in the rooting operation (step e) should be dependent upon the signal-to-noise ratio. This is, in fact, what has been observed with the INTEL Process (2, 20), root factors of one-third and one-half being found best dependent upon the signal-to-noise ratio.

One final observation about the INTEL technique can be made from Figure 4e. From this figure it is observed that the INTEL process does not preserve the triangular time weighting applied in step b. This may or may not result in any significant distortion depending upon how successive triangularly weighted windows sum. This particular problem has been previously observed and some work at compensating for this distortion has been pursued (20).

2. Tests with Real Speech

From listening to the two utterances (described at the beginning of this section) processed by INTEL, the advantages of processing speech in noise (by INTEL) are apparent. Clearly, speech processed by the INTEL technique "sounds better." This has been described as enhanced "listenability." While there is some question as to whether there is an enhanced intelligibility through the use of the INTEL technique, there seems no question as to a perceived improvement in the signal-to-noise ratio.

A recent communication (28) in evaluating the INTEL technique in signal-to-noise ratios of -5 dB, 0 dB, +5 dB, and no-noise, found little intelligibility gain for INTEL processed versus unprocessed speech.

3. An INTEL Extension

Figure 4g, the second-order spectrum for speech in no noise and at a signal-to-noise ratio of 0 dB, clearly reveals an increased amplitude near the origin which results from the additive noise. In the INTEL technique described with Figure 4 and Section V-A-1, a suppression of this second-order spectral peak to zero was implemented as described in two recent reports (1,2). Looking at Figure 4g (no noise), however, reveals that even for normal speech this peak is non-zero. In fact, from an observation of several synthetically generated speech sounds without noise, it appears that the peak in the second-order spectrum near the origin is usually approximately twice the amplitude of the peak due to the formant frequency envelope.

Pursuing this observation, an experiment was carried out in which the INTEL technique (step h) was changed to suppress this low frequency peak (and other low frequency harmonics) by a factor which

causes the zero frequency peak to be twice the amplitude of the formant peak in the processed output. This is accomplished by calculating a factor in the second-order spectrum given by:

$$X = \frac{\text{amplitude of the zero frequency component}}{\text{amplitude of the maximum formant envelope component}}$$

and dividing each of the ten low frequency, second-order harmonics by half this factor.

Upon listening to the result in comparison with INTEL a small improvement in intelligibility, listenability, and naturalness seemed apparent.

The use of this modification to the INTEL technique has advantages other than (perhaps) enhanced intelligibility. First, it should result in less distortion of the type shown in Figure 4j (and described earlier) since the "gate" is less severe. (This fact was not verified with synthetic speech.) Second, it results in an overall system automatically compensated for a changing signal-to-noise ratio.

A method of suppressing the zero frequency peak in the second-order spectrum by fixed factors has been previously investigated with INTEL (20). Results have indicated that a suppression of the peak by about one-third worked best at a signal-to-noise ratio of 0 dB. From Figure 4g, it can be noted that this is in very close agreement with causing the zero-frequency peak to be twice the formant peak as described in the previous paragraph.

B. SPECTRAL SUBTRACTION METHOD

As previously discussed, one problem with the INTEL technique is that it requires four Fourier Transformations. A process which is intuitively appealing, appears similar in function to "gating" in the

INTEL technique, and requires only two Fourier Transformations is simple spectral subtraction.

The basic method implemented for spectral subtraction consists of the following sequence of five steps:

(a) Input 512 time samples;

(b) Apply a triangular window to the time samples;

(c) Perform a 512 point FFT;

(d) Estimate the average noise level from the magnitude of the spectrum above 2.5 KHz, subtract this level from the magnitude spectrum and zero all frequency components above 3 KHz; and

(e) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the result from steps b and c above are the same as for the INTEL technique and are illustrated in Figure 4. The results for steps d and e are illustrated in Figure 5 for: (1) no noise (left) and (2) a signal-to-noise ratio of 0dB (right).

1. Discussion

One problem which results when subtracting from the magnitude spectrum involves the action to be taken when a difference results in a magnitude of less than zero. The method employed in this work for the subtraction of a level from the magnitude of the spectrum (step d previous paragraph) is slightly more complex than simple subtraction. If the value of the magnitude of a component is greater than the noise level, the noise level is subtracted from the magnitude value. If the value of the magnitude of a component is less than or equal to the noise level, that magnitude is divided by two.

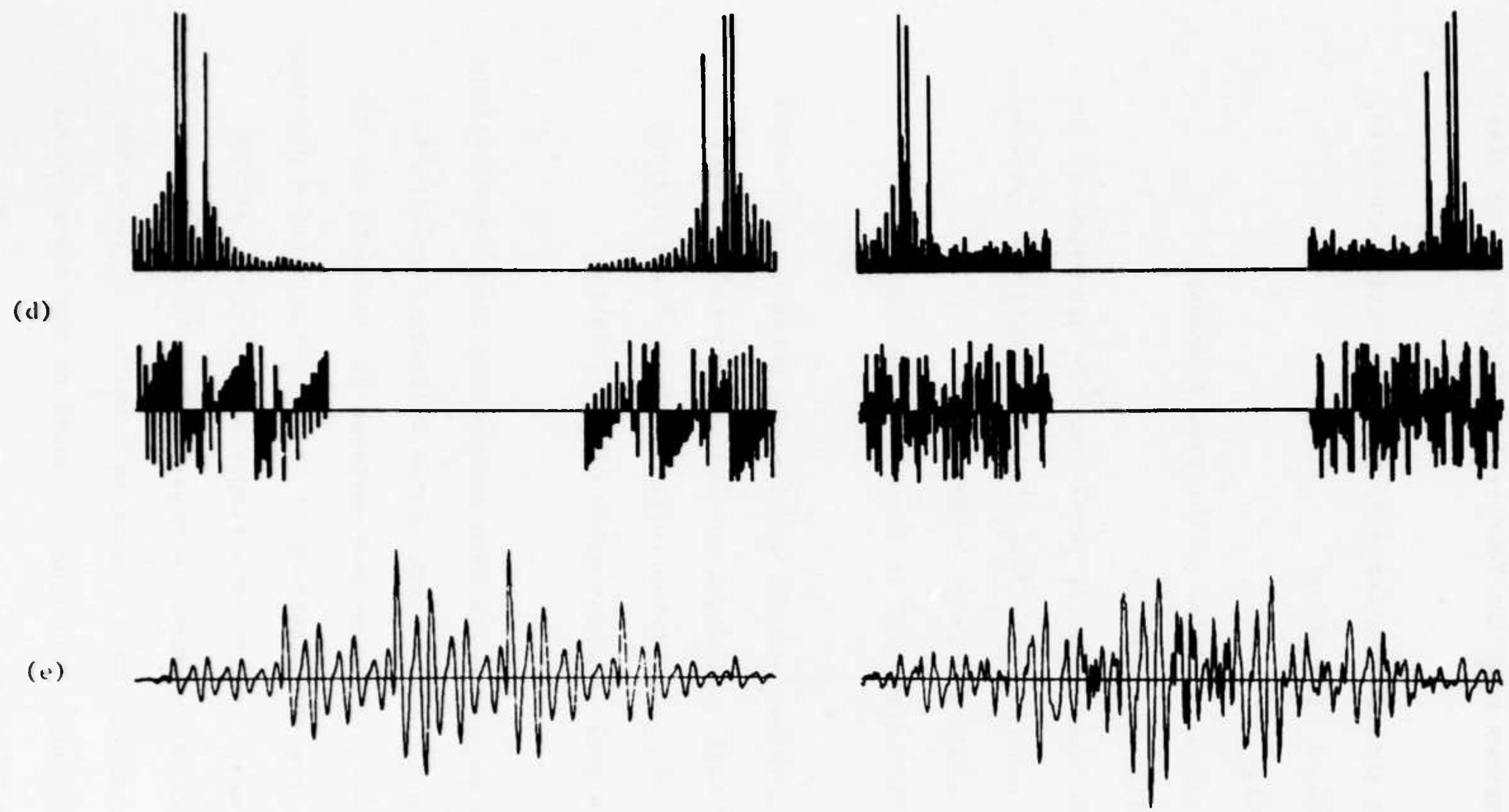


FIGURE 5. The spectral **subtraction** process applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-B; left is for a S/N = ∞, right is for a S/N = 0 dB.

From Figure 5d, it appears that the simple method employed to estimate and subtract the noise is not totally effective. Comparing Figure 5d with Figure 4k, the output spectrum from INTEL, reveals that INTEL appears to do a significantly better job at reducing noise than the subtractive method. Examining the time waveforms for the output from the spectral subtraction method (Figure 5e) and INTEL (Figure 4e) reveals that the spectral subtraction technique does not modify the time waveform as severely in the absence of noise. Probably a greater amount of subtraction (greater noise suppression) would be beneficial.

2. Tests With Real Speech

The results obtained for the processing of the real speech utterances described in the first paragraph of Section V confirm the expectations of the previous paragraph. The signal-to-noise ratio does not sound particularly enhanced over that obtained by simply zeroing the frequency range above 3 KHz and the intelligibility does not appear improved.

As a second experiment with subtractive noise cancellation, an additional processing step was added between steps d and e of the subtractive method. This additional process involves the use of a filter to emphasize the second formant frequency range. The characteristic of this filter was chosen to have a rising slope of 18 dB/octave below 1 KHz, a passband from 1 KHz to 2 KHz, and a falling slope of 12 dB/octave above 2 KHz. At a signal-to-noise ratio of -6 dB, the speech with high frequency emphasis sounds less intelligible than that with subtraction alone. This is probably due to the fact that enhancing the second formant frequency range is detrimental since (at this low signal-to-

noise ratio) the second formant range is so heavily obscured by noise. At a signal-to-noise ratio of +6 dB it is not clear whether the high frequency emphasis is helpful or not. Certainly, such an emphasis causes the speech to be less natural sounding in both cases.

C. MINIMUM MEAN-SQUARE-ERROR FILTERING

As indicated in Section II of this report, the method of minimum mean square error filtering is an attractive technique for processing speech in noise for two main reasons. First, it is an optimum method (in the least mean square error sense) for filtering a signal in noise; and second, it can be implemented in a computationally efficient manner (relative to other techniques).

The implementation used for this method is based upon an analysis from Papoulis (21) as described in Section II of this report. The implementation consists of the following sequence of five steps:

(a). Input 512 time samples;

(b). Apply a triangular window to the samples;

(c). Perform a 512 point FFT;

(d). Estimate the expected noise level from the magnitude of the spectrum above 2.5 KHz, estimate the expected signal from the long-time average for normal speech, and modify the magnitude spectrum using the filter H(jw) as given by Eq. 1 (Section II); and

(e). Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the results from steps b and c above are the same as those illustrated in Figure 4. The results for steps d and e are illustrated in Figure 6 for: (a) no noise (left) and

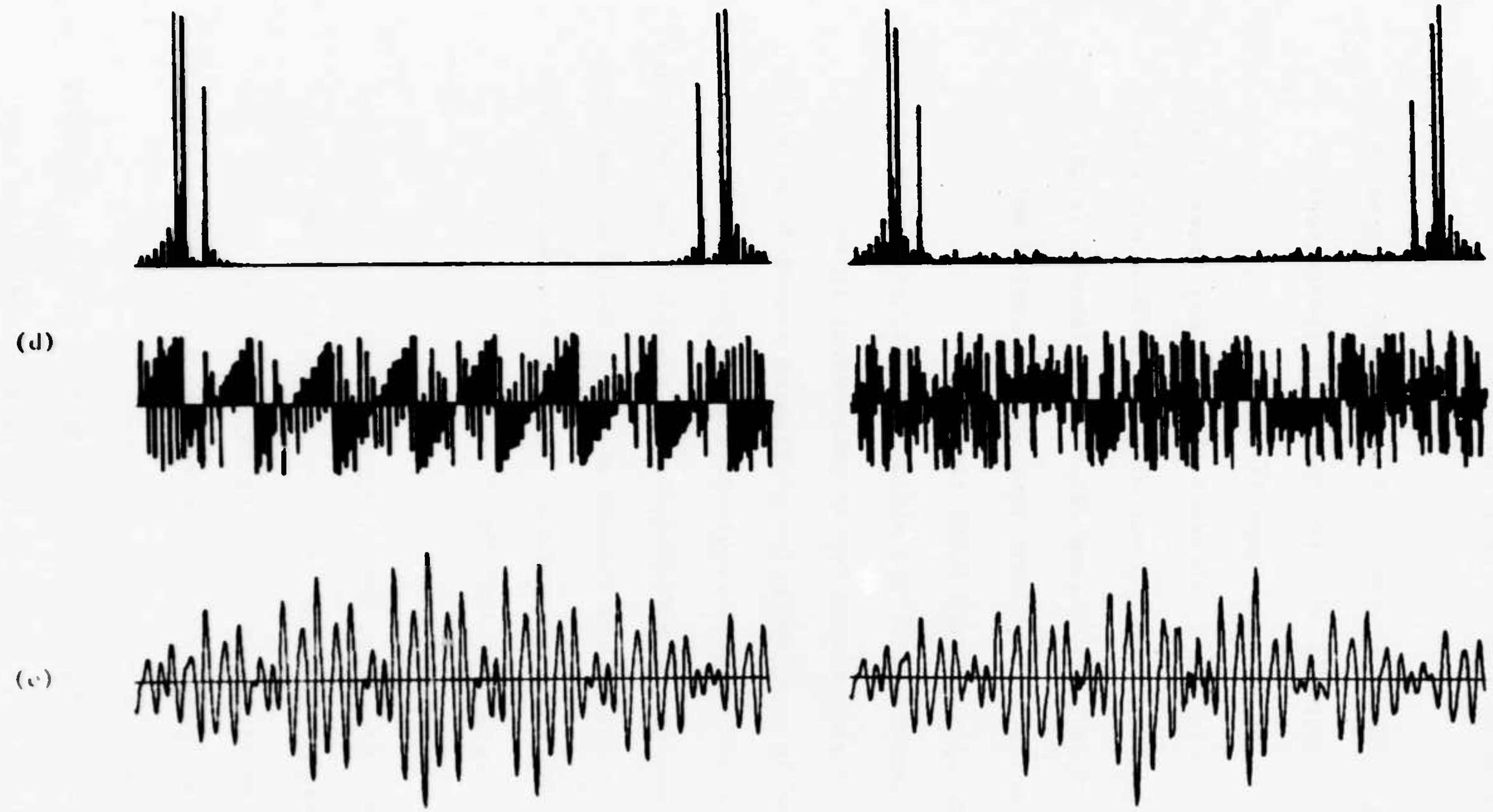


FIGURE 6. The minimum mean square error filtering process applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-C; left is for a S/N = ∞, right is for a S/N = 0 dB.

(2) a signal-to-noise ratio of 0 dB (right).

1. Discussion

A discussion of the minimum mean square error filtering process is contained in Section II of this report. A theoretical justification of Eq. 1 can be found in Papoulis (21).

From Figure 6 it can be observed that the magnitude of the frequency spectrum of the ouput (Figure 6d) with and without noise are surprisingly similar. The output spectrum after noise removal (right) appears very free from noise. The output time waveform after noise removal (Figure 6e) also appears very noise free.

One additional advantage of the minimum mean square error filtering method of processing speech for an enhancement in noise is that the method is directly extendible for processing speech in nonwhite noise environments. The only change required is a modification of Snn(w) in Eq. 1. In addition, if a technique were developed for detecting the presence of speech in noise, the absence of speech could be used to continually update a running noise estimate. This running noise estimate could then be readily incorporated into Eq. 1.

2. Tests With Real Speech

Like Figure 6 displays, the results achieved for processing real speech in noise (signal-to-noise ratios of +6 dB and -6 dB) by the method of minimum mean square error filtering are very encouraging. The signal-to-noise ratio is significantly enhanced, the naturalness unchanged, and the intelligibility sounds improved.

Two experiments were carried out with real speech, one using the procedure outlined in the second paragraph of this subsection (and

illustrated in Figure 6), the other with the addition of a zeroing of all magnitude components above 2.5 KHz between steps d and e. In both cases the results were very good with little observable difference between them. An examination of Figure 6d indicates that there is very little energy above 2.5 KHz such that little change would be expected.

For many reasons, the method of minimum mean square error filtering appears to offer a great potential for speech in noise intelligibility enhancement.

D. METHODS BASED UPON PITCH TRACKING

As described in Section II of this report, methods of enhancing the intelligibility of speech in noise based upon pitch analysis are intuitively attractive. However, as also indicated in Section II, such methods have, in general, produced discouraging results (20).

In order to experiment with methods based upon pitch tracking, a technique was implemented which consists of the following sequence of steps:

(a) Input 512 time samples;

(b) Apply a triangular window to the samples;

(c) Perform a 512 point FFT;

(d) Determine the pitch frequency using a method to be described and zero all magnitude components between pitch harmonics; and

(e) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the results from steps b and c above are the same as for the INTEL technique and are illustrated in Figure 4. The results after steps d and e are illustrated in Figure 7

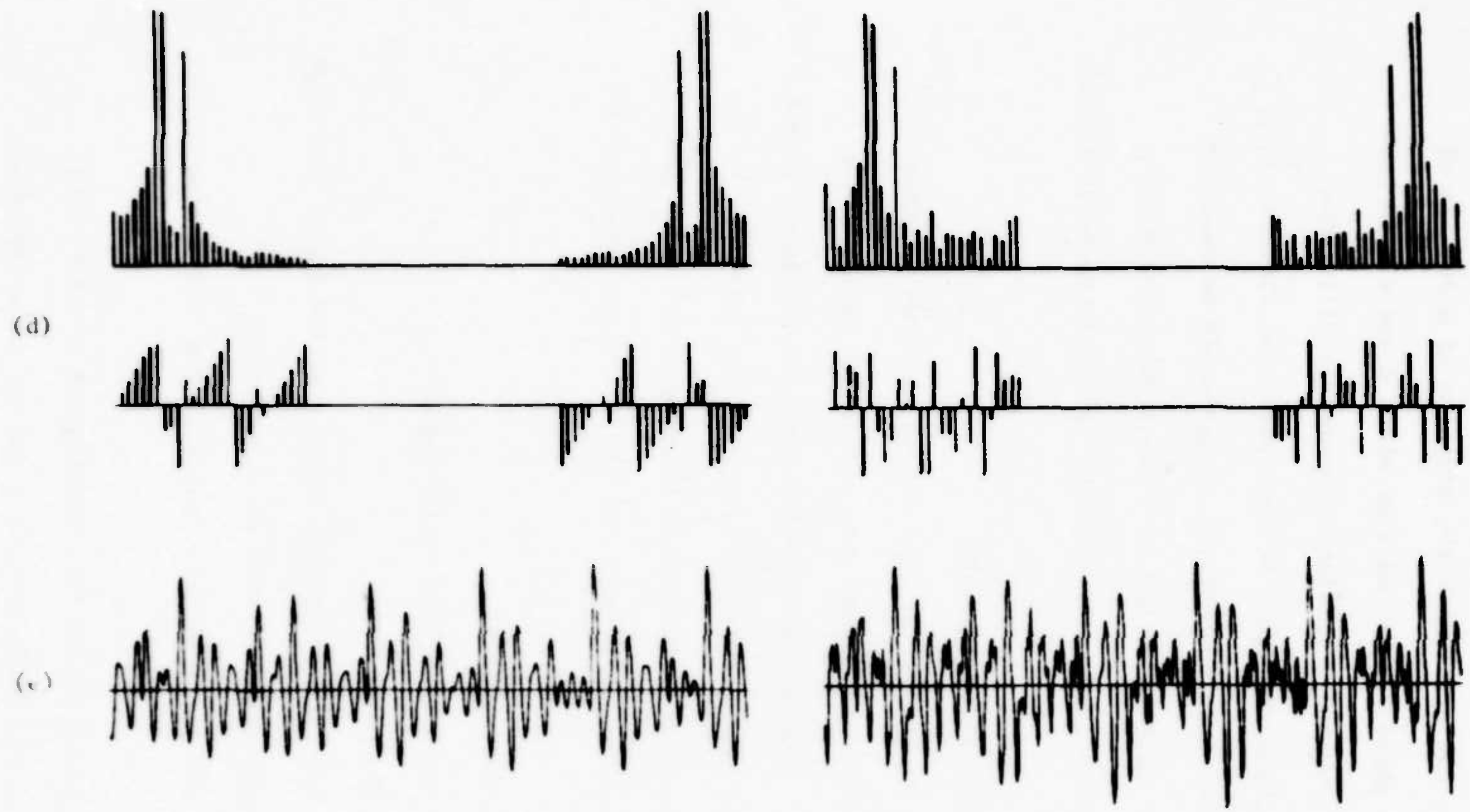


FIGURE 7. The process based upon pitch extraction applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-D; left is for a S/N = ∞, right is for a S/N = 0 dB.

for: (1) no noise (left) and (2) a signal-to-noise ratio of 0 dB (right).

1. Discussion

A discussion of the use of pitch extraction methods for speech intelligibility enhancement in noise is contained in Section II of this report.

The method used for estimating the pitch frequency from the spectrum of speech plus noise consists of determining that integer frequency (F_o) between 80 Hz and 250 Hz which maximizes the following function:

$$\frac{1}{N} \sum_{i=1}^{N} F(i \cdot F_0) \qquad \text{(Eq. 2)}$$

where:

N = the greatest integer such that $N \cdot F_o \leq 3000$ Hz

$F(i \cdot F_o)$ is that line of the magnitude spectrum closest to the frequency $i \cdot F_o$.

The results of Figure 7d display the expected spectrum of the output. With several tests on synthetic voiced sounds, at several signal-to-noise ratios from +6 dB to -6 dB, the pitch frequency was determined quite accurately (to within a few percent) using the method.

The output time waveform (Figure 7e) indicates the appearance of considerable improvement for the enhancement of speech in noise by using this method based upon pitch tracking.

2. Tests With Real Speech

Three experiments based upon pitch extraction and the method outlined earlier in this subsection were performed with real speech. These involve three degrees of suppression of non-pitch magnitude components in step d. These three degrees of suppression are: (1) suppression to zero, (2) suppression by a factor of two, and (3) suppression by a factor of four.

When the non-pitch components are suppressed to zero, a strong, low frequency distortion, probably a result of the time analysis window, is apparent. As the degree of suppression is lessoned the strong window distortion diminishes, however, at the expense of increased noise. At a high suppression of non-pitch harmonics (suppression to zero), there is a noticable loss of intelligibility, particularly at a signal-to-noise ratio of -6 dB. This is probably due to the inaccuracy in pitch tracking which is greater at lower signal-to-noise ratios. With a suppression of non-pitch harmonics to other than zero, the method is able to tolerate greater pitch tracking errors without as serious a degradation of speech intelligibility.

To determine some measure of the accuracy of the pitch tracking algorithm applied to real speech, two simple experiments were performed. First, a listing was created of the pitch values determined for the utterance at a signal-to-noise ratio of +6 dB. While no standard of comparison for the determined pitch values was available, the printed values seemed reasonable for a male speaker. The printed values were generally fairly continuous with fundamental frequency values between 110 Hz and 120 Hz during voiced speech intervals.

As a second experiment, a constant frequency of 10 Hz was subtracted from each pitch measurement and the result was used to generate the output waveform in the usual manner. Upon listening to the result it was found unintelligible.

Overall, the intelligibility results using this pitch tracking method sound of lesser intelligibility than either the INTEL or the minimum mean square error filtering methods. There are, however, a number of improvements which could be added to the basic method and which might substantially improve this first attempt. It is this author's feeling that methods based upon pitch tracking offer the potential to result in an enhancement of speech intelligibility in noise and that such methods should not be overlooked because of past discouraging results.

VI CONCLUSIONS

This study has explored several methods for the enhancement of speech intelligibility in the presence of wideband random noise at the speaker. This exploration has involved a study of the effect of the methods upon synthetically generated voiced sounds in noise as well as on real speech in white noise at two signal-to-noise ratios.

Four basic methods have been investigated: (a) INTEL, (b) spectral subtraction, (c) minimum mean square error filtering, and (d) methods based upon pitch tracking. Several variations in each basic method have been tested and numerous experiments with speech in noise have been performed. The experiments with synthetic speech have provided a substantial insight into not only the four methods, but also the speech in noise situation in general. Through the experiments with real speech, some qualitative results for the speech processing methods have been presented.

It appears from the experiments with synthetic and real speech that all four methods have the potential to result in an enhancement of the intelligibility of speech in noise. While the qualitative intelligibility results obtained during this work have indicated that, of the methods tested, the INTEL method and the method based upon minimum mean square error filtering seem to result in the greatest enhancement of speech in noise, the other methods should not be discarded. There are several reasons to suspect that substantial improvement in the intelligibility of speech in noise can be obtained through the use of each of the four basic techniques.

REFERENCES

1. M.R. Weiss, E. Aschkenasy, and T. W. Parsons, "Study and development of the INTEL Technique for Improving Speech Intelligibility," Technical Report No. RADC-TR-75-108, Rome Air Development Center, Griffiss Air Force Base, New York, April 1975. (AD# A011709)

2. M. R. Weiss and E. Aschkenasy, "Automatic Detection and Enhancement of Speech Signals," Technical Report No. RADC-TR-75-77, Rome Air Development Center, Griffiss Air Force Base, New York, March 1975.(AD# B004521L)

3. T.W. Parsons and M.R. Weiss, "Enhancing/Intelligibility of Speech in Noisy or Multi-talker Environments," Technical Report No. RADC-TR-75-155, Rome Air Development Center, Griffiss Air Force Base, New York, June 1975.(AD# A013767)

4. I.B. Thomas and A. Ravindran, "Intelligibility Enhancement of Already Noisey Speech Signals," J. Aud. Eng. Soc., 22, 234-236, 1974.

5. M.R. Sambur and N.S. Jayant, "LPC Analysis/Synthesis from Speech Inputs Containing Quantizing Noise or Additive White Noise," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-24, 488-494, 1976.

6. S.F. Boll, "Improving Linear Prediction Analysis of Noisy Speech by Predictive Noise Cancellation," 1977 IEEE International Conf. on Acoustics, Speech, and Signal Processing, 10-12, 1977.

7. R.J. Niederjohn, "A Comparison of Three Recently Reported Methods of Processing Speech for the Enhancement of Speech Intelligibility in High Noise Levels," Proceedings of the Midwest Symposium on Circuits and System, 325-331, 1976.

8. R.J. Niederjohn and J. H. Grotelueschen, "The Ehancement of Speech Intelligibility in High Noise Levels by High-pass Filtering Followed by Rapid Amplitude Compression," IEEE Trans. Acoust., Speech, and Sig. Proc., ASSP-24, 277-282, 1976.

9. R.J. Niederjohn and J. H. Grotelueschen, "Speech Intelligibility Enhancement in a Power Generating Noise Environment," IEEE Trans. Acoustics, Speech, and Signal Processing, accepted for publication.

10. I.B. Thomas and R.J. Niederjohn, "The Intelligibility of Filtered Clipped Speech in Noise," J. Aud. Eng. Soc., 18, 299-303, 1970.

11. I.B. Thomas and W.J. Ohley, "Intelligibility Enhancement Through Spectral Weighting," 1972 Conference on Speech Communication and Processing, 360-363, 1972.

12. J.P. Egan and F.M. Wiener, "On the Intelligibility of Bands of Speech in Noise," J. Acoust. Soc. Am., 18, 435-441, 1946.

13. I. Pollack and J.M. Pickett, "Masking of Speech by Noise at High Sound Levels," J. Acoust. Soc. Am., 30, 127-130, 1958.

14. I. Pollack and J.M. Pickett, "Intelligibility of Peak-Clipped Speech at High Noise Levels," J. Acoust. Soc., Am., 31, 14-16, 1959.

15. J.C.R. Licklider, "Effects of Amplitude Distortion on the Intelligibility of Speech," J. Acoust. Soc., Am., 18, 429-434, 1946.

16. G.A. Miller and S. Mitchell, "Effects of Distortion on the Intelligibility of Speech at High Altitudes," J. Acoust. Soc. Am., 19, 120-125, 1950.

17. W. Wathen-Dunn and D.W. Lipke, "On the Power Gained by Clipping Speech in the Audio Band," J. Acoust. Soc. Am., 30, 36-40, 1958

18. E.A. Kretsinger and N.B. Young, "The Use of Fast Limiting to Improve the Intelligibility of Speech in Noise," Speech Monogr., 27, 63-69, 1960.

19. H. Drucker, "Speech Processing in a High Ambient Noise Environment," IEEE Trans. Audio and Electroacoust., AU-16, 165-168, 1968.

20. Personal communication with Dr. Bruno Beek and Captain Robert Curtis, Rome Air Development Center, Griffiss Air Force Base, New York.

21. A.Papoulis, "Probability, Random Variables, and Stochastic Processes," McGraw-Hill Book Company, New York, 1965, pp 403-406.

22. G.E. Peterson and H.L. Barney, "Control Methods Used in a Study of Vowels," J. Acoust. Soc. Am. 24, 175-184. 1952.

23. N.R. French and J.C. Steinberg, "Factors Governing the Intelligibility of Speech Sounds," J. Acoust. Soc. Am., 19, 90-119, 1947.

24. H.K. Dunn and S.D. White, "Statistical Measurements on Conversational Speech," J. Acoust. Soc. Am., 11, 278-288, 1940.

25. I.B. Thomas, "The Second Formant and Speech Intelligibility," Proc. Nat. Elect. Conf., 23, 544-548, 1967.

26. I.B. Thomas, "The Influence of the First and Second Formants on the Intelligibility of Clipped Speech,: J. Audio Eng. Soc., 16, 182-185, 1968.

27. J.C.R. Licklider and I. Pollack, "Effects of Differentiation, Integration, and Infinite Peak Clipping Upon the Intelligibility of Speech," J. Acoust. Soc. Am., 20, 42-51, 1948.

28. Personal Communication from J.S. Lim, A. V. Oppenheim and L.D. Braida.

APPENDIX A

Source listing of the program "SPEECH", synthetic voiced speech generation program.

```
*LIST SPEECH

*#RUN#DATA"01"
         DIMENSION A(1024),VAR(3,3),W(3),AL(3)
        PI=3.1416
       PRINT,"INPUT: AMPL,FREQ,ALPHA, FOR ALL 3 FORMANTS"
        DO 20 I=1,3
 20     READ,(VAR(I,J),J=1,3)
        DO 30 I=1,3
        W(I)=2.*PI*VAR(I,2)
 30     AL(I)=2.*PI*VAR(I,3)
        DO 40 I=1,1024
        A(I)=0.
        T=I
        DO 40 J=1,3
 40     A(I)=A(I)+VAR(J,1)*EXP(-AL(J)*T/1.E4)*SIN(W(J)*T/1.E4)
       PRINT,"INPUT PITCH PERIOD IN MSEC"
       READ,FO
        ISHIFT=FO*10.
        DO 60 I=1,ISHIFT
        DO 60 II=I,1024,ISHIFT
 60     A(I)=A(I)+A(II)
        DO 65 I=ISHIFT+1,1024
 65     A(I)=A(I-ISHIFT)
        DO 70 I=1,512
 70     A(I)=A(I+512)
       PRINT,"NOISE?(1-YES, 0-NO)"
       READ,K
       IF(K)210,85,210
 210   PRINT,"INPUT S(RMS)/N(RMS) IN DB"
       READ,SN
       SPAMPL=0.
       DO 220 I=1.512
```

```
220   SPAMPL=SPAMPL+A(I)**2
      SPAMPL=SQRT(SPAMPL/512.)
      XNAMPL=SPAMPL/10.**(SN/20.)
      DO 230 I=1,512
230   A(I)=A(I)+XNAMPL*XNORM(1.)
85    PRINT,"VIEW ANY SAMPLES?(1-YES, 0-NO)"
       READ,K
       IF(K)90,110,90
90     WRITE(06,91)
91     FORMAT('SAMPLE NOS: FIRST, LAST')
       READ,K,L
       DO 100 I=K,L
100    WRITE(06,99)I,A(I)
99     FORMAT(1X,I4,4X,E17.10)
       GO TO 85
110    CALL RANSIZ(1,2,1)
       N=512;AA=0.
       WRITE(1'1)N,AA
       B=0.
       DO 120 J=2,513
120    WRITE(1'J)A(J),B
       CALL DETACH(01,ISTAT,)
       STOP
       END
      FUNCTION XNORM(X)
      X1=RAND(1.)
      X2=RAND(1.)
      Y=SQRT(-2.0*ALOG(X1))*(COS(6.2831853*X2))
      XNORM=Y
      RETURN
      END

ready
```

APPENDIX B

Source listing of the program "M2", synthetic speech processor. Only additions to the predecessor of this program "M1" (created by Mr. David Clark of RADC) are listed.

```
      .
      .
      .
60 GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500),M
      .
      .
      .
235C-----CALL TO INTEL PROCESSOR ON "11"
240 1100 CALL INTEL;GO TO 10
0245C---CALL TO RJN PROCESSOR ON "12"
0250 1200 CALL RJN(IMAX);GO TO 10
0251C---CALL TO SPECTRAL SUBTRACTION PROCESSOR
0252 1300 CALL SSP(XNOISE);GO TO 10
      .
      .
      .
0590C----INTEL PROCESSING ALGORITHM(PROG. BY R. NIEDERJOHN)
0600      SUBROUTINE INTEL
0610      COMMON ARRAY(63),DT,N,F(512,2),G(512,2)
0615      DIMENSION A(512,2)
0620      PRINT:"ENTER LAST STEP TO COMPLETE"
0630      READ:LAST
0631      PRINT:"ENTER STARTING POINT NUMBER"
0632      READ:NSTART;IF(NSTART.EQ.0)GO TO 1
0633      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NSTART
```

```
C----TRIANGULAR TIME WINDOW(1)
1     DO 19 J=1,256
      XJ=J
      F(J,1)=F(J,1)*XJ/256.
19    F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
      IF(LAST.EQ.1)RETURN
C----FOURIER TRANSFORM(2)
2     CALL CHAIN("LINK5")
      IF(LAST.EQ.2)RETURN
C----SAVE F IN A SO THAT PHASE MAY BE RESTORED LATER(3)
3     DO 20 I=1,512
      A(I,1)=F(I,1)
20    A(I,2)=F(I,2)
      IF(LAST.EQ.3)RETURN
C----MAKE RE(F) EQUAL TO PRESENT MAGNITUDE(4)
4     DO 30 I=1,512
30    F(I,1)=SQRT(F(I,1)**2+F(I,2)**2)
      IF(LAST.EQ.4)RETURN
C----SET IM(F) EQUAL TO ZERO(5)
5     DO 40 I=1,512
40    F(I,2)=0.
      IF(LAST.EQ.5)RETURN
C----ZERO UPPER HALF OF ARRAY(6)
6     DO 50 I=129,385
50    F(I,1)=0.
      IF(LAST.EQ.6)RETURN
C----TAKE SQRT OF ARRAY(7)
7     DO 60 I=1,512
60    F(I,1)=SQRT(F(I,1))
      IF(LAST.EQ.7)RETURN
C----REVERSE SIGN OF ALL ODD ARRAY ELEMENTS(8)
8     DO 70 I=1,511,2
70    F(I,1)=-F(I,1)
      IF(LAST.EQ.8)RETURN
C----TAKE SECOND FOURIER TRANSFORM(9)
9     CALL CHAIN("LINK5")
      IF(LAST.EQ.9)RETURN
C----SET LOW 5 ELEMENTS EQUAL TO ZERO(10)
```

```
1100 10   DO 100 I=253,261
1110      F(I,1)=0.
1115 100  F(I,2)=0.
1120      IF(LAST.EQ.10)RETURN
1130C----TAKE INVERSE FOURIER TRANSFORM(11)
1140 11   CALL CHAIN("LINK10")
1150      IF(LAST.EQ.11)RETURN
1160C----DISCARD IM(F) (12)
1170 12   DO 110 I=1,512
1180 110  F(I,2)=0.
1190      IF(LAST.EQ.12)RETURN
1200C----REVERSE SIGN OF ALL ODD ELEMENTS(13)
1210 13   DO 130 I=1,511,2
1220 130  F(I,1)=-F(I,1)
1230      IF(LAST.EQ.13)RETURN
1240C----SQUARE ARRAY(14)
1250 14   DO 140 I=1,512
1260 140  F(I,1)=F(I,1)**2
1270      IF(LAST.EQ.14)RETURN
1330C---RESTORE PHASE(15)
1340 15   DO 160 I=1,512
1350      F(I,2)=F(I,1)*SIN(ATAN2(A(I,2),A(I,1)))
1360 160  F(I,1)=F(I,1)*COS(ATAN2(A(I,2),A(I,1)))
1370      IF(LAST.EQ.15)RETURN
1380C---TAKE SECOND INVERSE FOURIER TRANSFORM(16)
1390 16   CALL CHAIN("LINK10")
1400      RETURN
1410      END
2000C----SPEECH IN NOISE PROCESSING ALGORITHM
2010      SUBROUTINE RJN(IMAX)
2020      COMMON ARRAY(63),DT,N,F(512,2),G(512,2)
2030      DIMENSION A(512,2)
2040      PRINT:"ENTER LAST STEP TO COMPLETE"
2050      READ:LAST
2060      PRINT:"ENTER STARTING POINT NUMBER"
2070      READ:NSTART:IF(NSTART.EQ.0)GO TO 1
2080      GO TO (1,2,3,4,5,6,7,8),NSTART
2090C----TRIANGULAR TIME WINDOW(1)
```

```
1     DO 19 J=1,256
      F(J,1)=F(J,1)*XJ/256.
19    F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
      IF(LAST.EQ.1)RETURN
C----FOURIER TRANSFORM(2)
2     CALL CHAIN("LINK5")
      IF(LAST.EQ.2)RETURN
C----SAVE F IN A FOR LATER RECONSTRUCTION(3)
3     DO 20 I=1,512
      A(I,1)=F(I,1)
20    A(I,2)=F(I,2)
      IF(LAST.EQ.3)RETURN
C----MAKE RE(F) EQUAL TO PRESENT MAGNITUDE(4)
4     DO 30 I=1,512
30    F(I,1)=SQRT(F(I,1)**2+F(I,2)**2)
      IF(LAST.EQ.4)RETURN
C----DETERMINE PITCH(5)
5     SUMMAX=0.
      DO 40 I=30,250
      XNUM=3000/I
      SUM=0.
      DO 50 J=I,3000,I
      XJ=J
      LNUM=XJ/19.53125+.5
      LNUM=LNUM+1
50    SUM=SUM+F(LNUM,1)
      SUM=SUM/XNUM
      IF(SUM.LE.SUMMAX)GO TO 40
      IF(IABS(IMAX-I/2).GE.5)GO TO 45
      IF(SUM.LE.1.33*SUMMAX)GO TO  40
45    SUMMAX=SUM
      IMAX=I
40    CONTINUE
      XIMAX=IMAX
      FO=1000./XIMAX
      PRINT:"PITCH FREQ.=",FO,"MSEC"
      IF(LAST.EQ.5)RETURN
C----CONSTRUCT NEW RE(F) &IM(F) KEEPING ONLY HARMONICS AT PITCH FREQ.(6)
```

```
6     DO 60 I=2.512
      F(I,1)=0.
60    F(I,2)=0
      DO 70 I=IMAX,3000,IMAX
      XI=I
      LNUM=XI/19.53125+.5
      LNUM=LNUM+1
      F(LNUM,1)=A(LNUM,1)
      F(LNUM,2)=A(LNUM,2)
      F(514-LNUM,1)=A(514-LNUM,1)
70    F(514-LNUM,2)=A(514-LNUM,2)
      IF(LAST.EQ.6)RETURN
C----INVERSE FOURIER TRANSFORM(7)
7     CALL CHAIN("LINK10")
      RETURN
C*****************************************************************************
C----CONSTRUCT NEW F KEEPING HARMONICS AT PITCH FREQ AND SUPPRESSING ALL
C    OTHERS BY 1/2.
8     DO 80 I=1.512
      F(I,1)=A(I,1)/2.
80    F(I,2)=A(I,2)/2.
      DO 85 I=154,360
      F(I,1)=0.
85    F(I,2)=0.
      DO 90 I=IMAX,3000,IMAX
      XI=I
      LNUM=XI/19.53125+.5
      LNUM=LNUM+1
      F(LNUM,1)=A(LNUM,1)
      F(LNUM,2)=A(LNUM,2)
      F(514-LNUM,1)=A(514-LNUM,1)
90    F(514-LNUM,2)=A(514-LNUM,2)
      IF(LAST.EQ.8)RETURN
C----INVERSE FOURIER TRANSFORM
      CALL CHAIN("LINK10")
      RETURN
      END
C----SPECTRAL NOISE SUBTRACTION PROCESSOR
```

```
      SUBROUTINE SSP(XNOISE)
      COMMON ARRAY(63),DT,N,F(512,2),G(512,2)
      DIMENSION A(512)
      PRINT:"ENTER LAST STEP TO COMPLETE"
      READ:LAST
      PRINT:"ENTER STARTING POINT NUMBER"
      READ:NSTART:IF(NSTART.EQ.0)GO TO 1
      GO TO (1,2,3,4,5,6,7,8),NSTART
C----TRIANGULAR TIME WINDOW(1)
1     DO 19 J=1,256
      XJ=J
      F(J,1)=F(J,1)*XJ/256.
19    F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
      IF(LAST.EQ.1)RETURN
C----FOURIER TRANSFORM(2)
2     CALL CHAIN("LINK5")
      IF(LAST.EQ.2)RETURN
C----ESTIMATE AVERAGE NOISE LEVEL(3)
3     SUM=0.
      DO 20 I=154,257
20    SUM=SUM+SQRT(F(I,1)**2+F(I,2)**2)
      XNOISE=SUM/104.
      IF(LAST.EQ.3)RETURN
C----MODIFY RE(F) AND IM(F) BY SUBTRACTING NOISE(4)
4     DO 30 I=1,512
      XMAG=SQRT(F(I,1)**2+F(I,2)**2)
      IF(XMAG.GT.2.*XNOISE)GO TO 40
      F(I,1)=F(I,1)/2.
      F(I,2)=F(I,2)/2.
      GO TO 30
40    F1=(XMAG-XNOISE)*COS(ATAN2(F(I,2),F(I,1)))
      F(I,2)=(XMAG-XNOISE)*SIN(ATAN2(F(I,2),F(I,1)))
      F(I,1)=F1
30    CONTINUE
      DO 50 I=154,360
      F(I,1)=0.
50    F(I,2)=0.
      IF(LAST.EQ.4)RETURN
C----INVERSE FOURIER TRANSFORM(5)
```

```
50    F(I,2)=0.
      IF(LAST.EQ.4)RETURN
C----INVERSE FOURIER TRANSFORM(5)
5     CALL CHAIN("LINK10")
      RETURN
C*****************************************************************************
C----MODIFY POWER SPECTRUM ACCORDING TO CURTIS' OPTIMUM METHOD(6)
6     DO 60 I=1,512
      RE=F(I,1)
      XIM=F(I,2)
      XMAG2=F(I,1)**2+F(I,2)**2
      XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
60    F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
      IF(LAST.EQ.6)RETURN
C----INVERSE FOURIER TRANSFORM
      CALL CHAIN("LINK10")
      RETURN
C*****************************************************************************
C----MODIFY POWER SPECTRUM AS IN 6 PLUS GATE OUT HIGH FREQ TERMS(7)
7     DO 70 I=1,512
      RE=F(I,1)
      XIM=F(I,2)
      XMAG2=F(I,1)**2+F(I,2)**2
      XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
70    F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
      DO 80 I=154,360
      F(I,1)=0.
80    F(I,2)=0.
      IF(LAST.EQ.7)RETURN
C----INVERSE FOURIER TRANSFORM
      CALL CHAIN("LINK10")
      RETURN
C*****************************************************************************
C----MODIFY POWER SPECTRUM AS IN 6 WITH WEIGHTING LIKE THAT FOR LONG TERM
C    AVERAGE OF NORMAL SPEECH
8     A(1)=0.$A(2)=1./2**8
```

```
      A(3)=1./2.**5
      A(4)=1./12.;A(5)=.5
      DO 90 I=6,27
90    A(I)=1.
      DO 100 I=28,257
      XI=I
      FREQ=(XI-1.)*19.53125
100   A(I)=(500./FREQ)**2
      SUM=0.
      DO 110 I=1,257
110   SUM=SUM+A(I)**2
      SUM=SQRT(SUM)/257.
      RNOISE=0.
      DO 120 I=154,257
120   RNOISE=RNOISE+SQRT(F(I,1)**2+F(I,2)**2)
      RNOISE=RNOISE/104.
      DO 130 I=1,257
130   A(I)=A(I)*RNOISE/SUM
      DO 140 I=2,256
140   A(514-I)=A(I)
      DO 150 I=1,512
      RE=F(I,1)
      XIM=F(I,2)
      XMAG2=F(I,1)**2+F(I,2)**2
      XMAG=(A(I)**2*XMAG2)/(A(I)**2+RNOISE**2)
      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
150   F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
      IF(LAST.EQ.8)RETURN
C----INVERSE FOURIER TRANSFORM
      CALL CHAIN("LINK10")
      RETURN
      END
```

APPENDIX C

Source listing of the program "RTB", magnetic tape read program. This program reads a record of length 512 words from tape with assigned number "10".

```
*LIST RTBFILE

010        SYMDEF   RTB
020        BLOCK    IO
030IBUF    BSS      512
040IEOF    BSS      1
050IBORT   BSS      1
060IPARTY  BSS      1
065        BSS      3328
070        USE      PREVIOUS
080RTB     SAVE
090        LDA      =-2,DL
100        STA      STR
110        MME      GEINOS
120        RTB
130        ZERO     FC,DCW
140        ZERO     STR
150        MME      GEROAD
160        LDA      STR
170        ANA      =007000000000
180        CMPA     =003000000000
190        TZE      ABORT
200        CMPA     =004000000000
210        TZE      EOF
220        LDA      =0,DL
230        TRA      RETURN
240ABORT   LDA      =1,DL
250        STA      IPARTY
260        TRA      RETURN+1
270EOF     LDA      =1,DL
280RETURN  STA      IEOF
```

```
290         RETURN   RTB
300FC       BCI      1,000010
310 CH      IOTD     IBUF,512
320STR      BSS      2
330         END

ready
```

APPENDIX D

Source listing of the program "WTB", magnetic tape write program. This program writes a record of length 512 words onto tape with assigned number 20.

```
*LIST WTBFILE

0010        SYMDEF   WTB
0020        BLOCK    IO
0030IBUF    BSS      512
0040IEOF    BSS      1
0050ISORT   BSS      1
0060IPARTY  BSS      1
0070        BSS      3328
0080        USE      PREVIOUS
0090WTB     SAVE
0100        MME      GEINOS
0110        WTB
0120        ZERO     FC,DCW
0130        ZERO     STR
0140        MME      GEROAD
0150        RETURN   WTB
0160FC      BCI      1,000020
0170DCW     IOTD     IBUF,512
0180STR     BSS      2
0190        END

ready
```

APPENDIX E

Source listing of the program "WTBZ", magnetic tape write program. This program writes a record of length 1 word onto tape with assigned number 20

```
*LIST WTBZFILE

0010        SYMDEF   WTBZ
0020        BLOCK    IO
0030IBUF    BSS      512
0040IEOF    BSS      1
0050IBORT   BSS      1
0060IPARTY  BSS      1
0070        BSS      3328
0080        USE      PREVIOUS
0090WTBZ    SAVE
0100        MME      GEINOS
0110        WTB
0120        ZERO     FC,DCW
0130        ZERO     STR
0140        MME      GEROAD
0150        RETURN   WTBZ
0160FC      BCI      1,000020
0170DCW     IOTD     IBUF,1
0180STR     BSS      2
0190        END

ready
```

APPENDIX F

Batchjob listing of the sequence of operations to be performed in copying one magnetic tape (assigned name "IN") onto another magnetic tape (assigned name "OT") which may be read without parity errors.

```
*LIST UTILITY1

0020$:IDENT:AIAD0005,NIEDERJOH,40270001RADC
0022$:USERID:AIAD0005$WORLD
0025$:MSG1:NICK_DO A "KILL" AFTER EOJ-NIEDERJOHN
0030$:UTILITY
0040$:QUTIL:IGNORE/145
0050$:FFILE:IN,PHYREC
0060$:FUTIL:IN,OT,RWD/IN,OT/,COPY/144R/
0070$:TAPE7:IN,A1D,,S0621,,INPUT,,DEN5
0080$:TAPE7:OT,X2D,,S0622,,OUTPUT,,DEN5
0090$:LIMITS:99,20K,,10000
0100$:ENDJOB

ready
```

APPENDIX G

Batchjob and source listing of a program (RTAPE) to copy a magnetic tape to a data file. Note that program "RTBFILE" (Appendix C) is used.

```
*LIST BATCHJOB

$:IDENT:AIAD0005,NIEDERJOH,40270001RADC
$:USERID:AIAD0005$WORLD
$:OPTION:FORTRAN
$:FORTY:NDECK
$:SELECTA:AIAD0005/RTAPE
$:GMAP:NDECK
$:SELECTA:AIAD0005/RTBFILE
$:EXECUTE
$:LIMITS:25,25K
$:PRMFL:12,R/W,L,AIAD0005/TAPDAT2
$:TAPE7:10,X1D,,50623,,SPCH + NOISE,,DEN5
$:FFILE:10,NSTDLB,BUFSIZ/512,NOSRLS
$:ENDJOB

ready

*LIST RTAPE

      COMMON /IO/ IBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
      CALL RTB;CALL RTB;CALL RTB
      DO 1 I=1,150
      CALL RTB
      IF(IEOF.NE.0)GO TO 2
      WRITE(12)IBUF
    1 CONTINUE
    2 CONTINUE
      STOP
      END

ready
```

APPENDIX H

Batchjob (FITA25) and source listing for a program (PROCESS) to process real speech. Input data is from a data file (TAPDAT2) and output data is directed to tape with label "SO625". Note that programs "WTBFILE" (Appendix D) and "WTBZFILE" (Appendix E) are called during execution of this job.

```
*LIST FITA25

$:IDENT:AIAD0005,NIEDERJOH,40270001RADC
$:USERID:AIAD0005$WORLD
$:OPTION:FORTRAN
$:FORTY:NDECK
$:LIMITS:,30K
$:SELECTA:AIAD0005/PROCESS
$:GMAP:NDECK
$:SELECTA:AIAD0005/WTBFILE
$:GMAP:NDECK
$:SELECTA:AIAD0005/WTBZFILE
$:EXECUTE
$:LIMITS:100,25K
$:PRMFL:10,R/W,L,AIAD0005/TAPDAT2
$:TAPE7:20,X20,,S0625,,OUTPUT,,DEN5
$:FFILE:20,NSTDLB,BUFSIZ/512,NOSRLS
$:ENDJOB

ready

*LIST PROCESS

C----ESTABLISH ARRAYS USED BY THE VARIOUS PROCESSING ALGORITHMS:
C             IOBUF(512)-FOR I/O FROM/TO TAPE
C             IEOF,IBORT,IPARTY-READ ERROR FLAGS
C             IDATA(2048)-PAST AND PRESENT DATA RECORDS
C             JODATA(1280)-OUTPUT DATA RECORD
C             F(512,2)-PROCESSING BUFFER
```

```
C            G(512,2)-TEMP BUFFER USED BY FFT AND IFFT
C            A(512,2)-TEMP BUFFER USED IN THE ALGORITHMS
C            S(512)-SIN/COS TABLE USED BY FFT & IFFT
      COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      COMMON /SCTAB/ S(512)
C----NPROC SHOULD EQUIL THE NUMBER OF ENHANCEMENT PROCESSES TO BE
C     PERFORMED
      NPROC=4
C----NREC SHOULD EQUIL THE NUMBER OF RECORDS TO BE PROCESSED BY EACH
C      ENHANCEMENT PROCESS, NREC+NFREC MUST BE .LE. TOT. NUM. REC. INPUT
      NREC=120
C----NFREC SHOULD EQUIL THE FIRST RECORD TO BE PROCESSED
      NFREC=1
      REWIND 10;REWIND 20;CALL TABLF(512)
C----WRITE A ONE WORD RECORD FOR THE DDP-116
      DO 50 I=1,1280
 50   JODATA(I)=0
      CALL DATTAP
      CALL WTBZ
C----FIRST JUST COPY INPUT TO OUTPUT
      DO 62 I=1,NFREC-1
 62   READ(10)IOBUF
      DO 51 I=1,NREC
      CALL PTIME(A1);READ(10)IOBUF
 51   CALL WTB;CALL PTIME(B)
C----WRITE TEN ZERO RECORDS
      DO 52 I=1,1280
 52   JODATA(I)=0
      CALL DATTAP;DO 55 I=1,10
 55   CALL WTB;REWIND 10
      JR1=1;JR2=NREC
      WRITE(06,53)JR1,JR2
 53   FORMAT(1X,"INPUT SPEECH,           REC. NOS. ",I5," TO ",I5)
      TI=B-A1
      WRITE(06,54)TI
 54   FORMAT(10X,"PROCESSING TIME= ",F10.6," HOURS")
C----BEGIN PROCESSING TECHNIQUES
```

```
      DO 400 J=1,NPROC
      DO 60 I=1,1280
60    JODATA(I)=0
C----JJ IS RECORD COUNTER
      DO 61 JJ=1,NFREC
61    READ(10)IOBUF
      CALL TAPDAT(1)
      CALL PTIME(A1)
      DO 300 JJ=1,NREC
      READ(10)IOBUF
      CALL TAPDAT(2)
C----I IS INTERRECORD SEGMENT COUNTER
      DO 200 I=1,4
      N=(I-1)*256+1
      DO 70 II=N,N+511
      F(II-N+1,1)=IDATA(II)
      F(II-N+1,1)=F(II-N+1,1)/1000.
70    F(II-N+1,2)=0.
C----CALLS TO SUCCESSIVE PROCESSING TECHNIQUES
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,13,19,20),J
1     CALL INTEL;GO TO 49
2     CALL SUBT1;GO TO 49
3     CALL SUBT2;GO TO 49
4     CALL PAP;GO TO 49
5     CALL MODPAP;GO TO 49
6     CALL SPPAP;GO TO 49
7     CALL MODSPPAP;GO TO 49
8     CALL RJN1;GO TO 49
9     CALL RJN2;GO TO 49
10    CALL RJN3;GO TO 49
11    CALL RJN4;GO TO 49
12    CONTINUE
13    CONTINUE
14    CONTINUE
15    CONTINUE
16    CONTINUE
17    CONTINUE
18    CONTINUE
19    CONTINUE
```

```
20    CONTINUE
49    CONTINUE
C----NORMALIZE OUTPUT
      II=I;JJJ=JJ
      CALL NORM1(II,JJJ,SFACT)
79    DO 80 II=N,N+511
      FJO=F(II-N+1,1)*SFACT
      JO=SIGN(1.0,FJO)*(ABS(FJO)+.5)
80    JODATA(II)=JODATA(II)+JO
200   CONTINUE
C----OUTPUT PROCESSED RECORD
      GO TO 202
C----TEMP TEST
      IF(JJ.GT.2)GO TO 202
      WRITE(06,201)JODATA
201   FORMAT(1X,10(I8,2X))
      WRITE(06,203)J,JJ
203   FORMAT(20X,"PROCESS ",I3,", RECORD ",I3)
202   CONTINUE
C----TEMP TEST
      CALL DATTAP
      CALL WTB
C----REINITIALIZE JODATA & IDATA
      DO 210 I=1,256
210   JODATA(I)=JODATA(I+1024)
      DO 220 I=257,1280
220   JODATA(I)=0
      DO 230 I=1,1024
230   IDATA(I)=IDATA(I+1024)
300   CONTINUE
C----OUTPUT TEN ZERO RECORDS
      CALL PTIME(B)
      DO 390 I=1,1024
390   JODATA(I)=0
      CALL DATTAP
      DO 391 I=1,10
391   CALL WTB
      REWIND 10
```

```
      JR1=J*(NREC+10)+1
      JR2=JR1+NREC-1
      WRITE(06,399)J,JR1,JR2
  399 FORMAT(1X,"PROCESS",2X,I3,2X,"COMPLETE, REC. NOS. ",I5," TO ",I5)
      TI=B-A1
      WRITE(06,54)TI
  400 CONTINUE
      STOP
      END
C************************************************************************
C--PROCESSING ALGORITHMS
C       INPUT/OUTPUT FOR ALGORITHMS IS F
C************************************************************************
      SUBROUTINE INTEL
C----INTEL PROCESSOR
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL ITW
      CALL FFT
      CALL FTOA
      CALL ZIMF
      CALL ZUHALF
      CALL SQRTF
      CALL REVSIGN
      CALL FFT
      CALL ZLOW
      CALL IFFT
      CALL ZIMF
      CALL REVSIGN
      CALL SQUARE
      CALL RESTPH
      CALL IFFT
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE SUBT1
C----SUBTRACT AVERAGE NOISE FROM FFT(F)
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL ITW
```

```
      CALL FFT
      CALL ESTNOISE(XNOISE)
      CALL NOISESUB(XNOISE)
      CALL IFFT
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE SUBT2
C----SUBTRACT AVERAGE NOISE, THEN WEIGHT TO EMPHASIZE F2
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL ESTNOISE(XNOISE)
      CALL NOISESUB(XNOISE)
      CALL WEIGHT3
      CALL IFFT
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE PAP
C----POWER SPECTRAL OPTIMIZATION WITH ACTUAL SPEECH AS EST. SPEECH
C    SPECTRUM
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL ESTNOISE(XNOISE)
      CALL WEIGH1(XNOISE)
      CALL IFFT
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE MODPAP
C----SAME AS PAP WITH UPPER HALF OF FFT SET TO ZERO
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL ESTNOISE(XNOISE)
      CALL WEIGH1(XNOISE)
```

```
      CALL ZUHALF
      CALL IFFT
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE SPPAP
C----POWER SPECTRAL OPTIMIZATION WITH LONG TERM AVERAGE SPECTRUM USED AS
C    ESTIMATED SPEECH SPECTRUM
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL WEIGH2
      CALL IFFT
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE MODSPPAP
C----SAME AS SPPAP WITH UPPER HALF OF FFT SET TO ZERO
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL WEIGH2
      CALL ZUHALF
      CALL IFFT
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE RJNI
C----LEAVE SPECTRAL LINES AT PITCH FREQ. ONLY
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FTOA
      CALL MAG
      CALL PITCH(X)
      CALL HARM(X,O.)
      CALL IFFT
      RETURN
```

```
      END
C------------------------------------------------------------------------
      SUBROUTINE RJN2
C----LEAVE SPECTRAL LINES AT PITCH FREQ. WITH ALL OTHERS SUPPRESSED
C    BY 1/2.
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FTOA
      CALL MAG
      CALL PITCH(X)
      X=X-10.
      CALL HARM(X,.5)
      CALL IFFT
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE RJN3
C----LEAVE SPECTRAL LINES AT PITCH FREQ. WITH ALL OTHERS SUPPRESSED
C    BY 1/4.
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FTOA
      CALL MAG
      CALL PITCH(X)
      CALL HARM(X,.25)
      CALL IFFT
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE RJN4
C----SAME AS RJN1 WITH WEIGHTING ON F LIKE THAT CONSIDERING THE
C    IMPORTANCE OF EACH LINE TO NORMAL SPEECH
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FTOA
```

```
      CALL MAG
      CALL PITCH(X)
      CALL HARM(X,0.)
      CALL WEIGHT3
      CALL IFFT
      RETURN
      END
C---------------------------------------------------------------------
      SUBROUTINE NOPROC
C----NO PROCESS,JUST TEST
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TIM
      RETURN
      END
C---------------------------------------------------------------------
      SUBROUTINE INTEL2
C----MODIFIED INTEL PROCESS
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TIM
      CALL FFT
      CALL FTOA
      CALL MAG
      CALL ZIMF
      CALL ZUHALF
      CALL SQRTF
      CALL REVSIGN
      CALL FFT
      CALL ZLOW2
      CALL IFFT
      CALL ZIMF
      CALL REVSIGN
      CALL SQUARE
      CALL RESTPH
      CALL IFFT
      RETURN
      END
C---------------------------------------------------------------------
      SUBROUTINE INTEL3
```

```
C----INTEL WITHOUT SQUARE
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FTOA
      CALL MAG
      CALL ZIMF
      CALL ZUHALF
      CALL SORTF
      CALL REVSIGN
      CALL FFT
      CALL ZLOW
      CALL IFFT
      CALL ZIMF
      CALL REVSIGN
      CALL ABSVF
      CALL RESTPH
      CALL IFFT
      RETURN
      END
C--------------------------------------------------------------------
      SUBROUTINE SUBT3
C----SUBTRACT NOISE FROM FFT
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL ESTNOISE(XNOISE)
      CALL NOISUR2(XNOISE)
      CALL IFFT
      RETURN
      END
C********************************************************************
C--SUBROUTINES USED BY THE VARIOUS PROCESSING ALGORITHIMS
C********************************************************************
      SUBROUTINE TTW
C----TRIANGULAR TIME WINDOW ON F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 10 I=1,256
```

```
      XI=I-1
      F(I,1)=F(I,1)*XI/255.
   10 F(I+256,1)=F(I+256,1)*(255.-XI)/255.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE FFT
C----FOURIER TRANSFORM
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      COMMON /SCTAB/ S(512);N=512
      DO 100 I=1,N;NN=N;II=I
      CALL FILE(NN,II,J)
      G(J,1)=F(II,1)
      G(J,2)=F(II,2)
  100 CONTINUE
      CALL FORIER(N,G)
      DO 200 I=1,N
      F(I,1)=G(I,1)
      F(I,2)=G(I,2)
  200 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE FILE(N,J,I)
C----SUBROUTINE USED BY FFT AND IFFT
      DIMENSION L(12),LA(12)
      I=1
      IF(J.NE.1)GO TO 103
      DO 100 K=1,12
      LA(K)=0
      NA=2**K
      L(K)=N/NA
  100 IF(NA.EQ.N)GO TO 101
  101 CONTINUE
      M=K+1
      IF(K.EQ.12)GO TO 110
      DO 102 K=M,12
      LA(K)=0
```

```
      L(K)=0
  102 CONTINUE
  110 CONTINUE
      RETURN
  103 DO 104 K=1,12
      IF(LA(K).NE.0)GO TO 104
      LA(K)=1
      GO TO 105
  104 LA(K)=0
  105 DO 106 K=1,12
  106 I=I+LA(K)*L(K)
      RETURN
      END
C---------------------------------------------------------------------
      SUBROUTINE FORIER(N,F)
C----SUBROUTINE USED BY FFT
      COMMON /SCTAB/ S(512)
      DIMENSION F(512,2)
      DO 100 I=1,10
      K=2**I
  100 IF(N.EQ.K)GO TO 101
      STOP
C-----ERROR THAT N IS NOT MOD 2 OR N IS GREATER THAN 1024 OR LESS THAN 2
  101 K=I
      K1=2**(K-2)
      K2=2*K1
      KSTP=2*K1-1
      DO 102 I=1,K
      L=2**(I-1)
      I1=2*L
      DO 104 I2=1,N,I1
      DO 103 J=1,L
      M=I2+J-1
      AR=F(M,1)
      AI=F(M,2)
      BR=F(M+L,1)
      BI=F(M+L,2)
      MA=(J-1)*K2/L+1
```

```
      CI=-S(MA)
      IF(MA-KI)107,107,105
  107 KK=MA+KI
      CR=S(KK)
      GO TO 106
  105 KK=MA-KI
      CR=-S(KK)
  106 DR=BR*CR-BI*CI
      DI=BR*CI+BI*CR
      F(M,1)=AR+DR
      F(M,2)=AI+DI
      F(M+L,1)=AR-DR
      F(M+L,2)=AI-DI
  103 CONTINUE
  104 CONTINUE
  102 CONTINUE
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE FTOA
C----SAVE F IN A
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 20 I=1,512
      A(I,1)=F(I,1)
   20 A(I,2)=F(I,2)
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE MAG
C----MAKE RE(F) EQUAL TO PRESENT MAGNITUDE
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 30 I=1,512
   30 F(I,1)=SQRT(F(I,1)**2+F(I,2)**2)
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE ZIMF
C----SET IM(F)=0
```

```
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 40 I=1,512
   40 F(I,2)=0.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE ZUHALF
C----ZERO UPPER HALF OF F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 50 I=129,385
      F(I,1)=0.
   50 F(I,2)=0.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE SQRTF
C----TAKE SQRT OF F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 60 I=1,512
   60 F(I,1)=SQRT(F(I,1))
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE REVSIGN
C----REVERSE SIGNS OF ALL ODD ELEMENTS OF F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 70 I=1,511,2
   70 F(I,1)=-F(I,1)
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE ZLOW
C----ZERO LOW FIVE ELEMENTS OF F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 100 I=253,261
      F(I,1)=0.
  100 F(I,2)=0.
      RETURN
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE IFFT
C----INVERSE FOURIER TRANSFORM
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      COMMON /SCTAB/ S(512)
      N=512
      DO 100 I=1,N;NN=N;II=I
      CALL FILE(NN,II,J)
      G(J,1)=F(II,1)
      G(J,2)=F(II,2)
  100 CONTINUE
      CALL IFORIER(N,G)
      DO 200 I=1,N
      F(I,1)=G(I,1)
      F(I,2)=G(I,2)
  200 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE IFORIER(N,F)
C----SUBROUTINE USED BY IFFT
      COMMON /SCTAB/ S(512)
      DIMENSION F(512,2)
      DO 100 I=1,10
      K=2**I
  100 IF(N.EQ.K)GO TO 101
      STOP
C-----ERROR THAT N IS NOT MOD 2 OR N IS GREATER THAN 1024 OR LESS THAN 2
  101 K=I
      K1=2**(K-2)
      K2=2*K1
      KSTP=2*K1-1
      DO 102 I=1,K
      L=2**(I-1)
      I1=2*L
      DO 104 I2=1,N,I1
      DO 103 J=1,L
      M=I2+J-1
```

```
      AR=F(M,1)
      AI=F(M,2)
      BR=F(M+L,1)
      BI=F(M+L,2)
      MA=(J-1)*K2/L+1
      CI=S(MA)
      IF(MA-K1)107,107,105
 107  KK=MA+K1
      CR=S(KK)
      GO TO 106
 105  KK=MA-K1
      CR=-S(KK)
 106  DR=BR*CR-BI*CI
      DI=BR*CI+BI*CR
      F(M,1)=AR+DR
      F(M,2)=AI+DI
      F(M+L,1)=AR-DR
      F(M+L,2)=AI-DI
 103  CONTINUE
 104  CONTINUE
 102  CONTINUE
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE SQUARE
C----SQUARE RE(F)
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 100 I=1,512
 100  F(I,1)=F(I,1)**2
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE RESTPH
C----RESTORE PHASE OF F FROM A
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 10 I=1,512
      F(I,2)=F(I,1)*SIN(ATAN2(A(I,2),A(I,1)))
 10   F(I,1)=F(I,1)*COS(ATAN2(A(I,2),A(I,1)))
```

```
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE ESTNOISE(XNOISE)
C----ESTIMATE AVERAGE NOISE LEVEL
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      SUM=0.
      DO 20 I=154,257
   20 SUM=SUM+SQRT(F(I,1)**2+F(I,2)**2)
      XNOISE=SUM/104.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE NOISESUB(XNOISE)
C----MODIFY F BY SUBTRACTING NOISE AND SET F(I,1 AND 2)=0.,I=154,360
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 30 I=1,512
      XMAG=SQRT(F(I,1)**2+F(I,2)**2)
      IF(XMAG.GT.2.*XNOISE)GO TO 40
      F(I,1)=F(I,1)/2.
      F(I,2)=F(I,2)/2.
      GO TO 30
   40 F1=(XMAG-XNOISE)*COS(ATAN2(F(I,2),F(I,1)))
      F(I,2)=(XMAG-XNOISE)*SIN(ATAN2(F(I,2),F(I,1)))
      F(I,1)=F1
   30 CONTINUE
      DO 50 I=154,360
      F(I,1)=0.
   50 F(I,2)=0.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE WEIGHT(XNOISE)
C----MODIFY POWER SPECTRUM ACCORDING TO CURTIS' OPTIMUM METHOD
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 60 I=1,512
      RE=F(I,1)
      XIM=F(I,2)
```

```
      IF((XIM.EQ.0.).AND.(RE.EQ.0.))GO TO 60
      XMAG2=F(I,1)**2+F(I,2)**2
      XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
      F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
 60   CONTINUE
      RETURN
      END
C---------------------------------------------------------------------------
      SUBROUTINE WEIGH2
C----MODIFY POWER SPECTRUM AS IN WEIGHT1 BUT WITH WEIGHTING LIKE THAT FOR
C    LONG TERM AVERAGE OF NORMAL SPEECH
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
C----CALCULATE ARRAY G, THE LONG TERM AV. FOR NORMAL SPEECH
      G(1,1)=0.;G(2,1)=1./2.**8;G(3,1)=1./2.**5;G(4,1)=1./12.;G(5,1)=.5
      DO 90 I=6,27
 90   G(I,1)=1.
      DO 100 I=28,257
      XI=I
      FREQ=(XI-1.)*19.53125
 100  G(I,1)=(500./FREQ)**2
      SUM=0.
      DO 110 I=1,257
 110  SUM=SUM+G(I,1)
      SUM=SUM/257.
C----CALCULATE AVERAGE OF MAG(F)
      FSUM=0.
      DO 115 I=1,257
 115  FSUM=FSUM+SQRT(F(I,1)**2+F(I,2)**2)
      FSUM=FSUM/257.
C----CALCULATE NOISE AVERAGE
      RNOISE=0.
      DO 120 I=154,257
 120  RNOISE=RNOISE+SQRT(F(I,1)**2+F(I,2)**2)
      RNOISE=RNOISE/104.
      DO 130 I=1,257
 130  G(I,1)=G(I,1)*FSUM/SUM
      DO 140 I=2,256
 140  G(514-I,1)=G(I,1)
```

```
      DO 150 I=1.512
      RE=F(I.1)
      XIM=F(I.2)
      XMAG2=F(I.1)**2+F(I.2)**2
      XMAG=(G(I.1)**2*XMAG2)/(G(I.1)**2+RNOISE**2)
      F(I.1)=XMAG*COS(ATAN2(XIM.RE))
  150 F(I.2)=XMAG*SIN(ATAN2(XIM.RE))
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE PITCH(X)
C----CALCULATE PITCH AS HARMONIC NUMBER = X
      COMMON /BUFFER2/ F(512.2).G(512.2).A(512.2)
      SUMMAX=0.
      DO 40 I=80.250
      XNUM=3000/I
      SUM=0.
      DO 50 J=I.3000.I
      XJ=J
      LNUM=XJ/19.53125+.5
      LNUM=LNUM+1
   50 SUM=SUM+F(LNUM.1)
      SUM=SUM/XNUM
      IF(SUM.LE.SUMMAX)GO TO 40
      IF(IABS(IMAX-I/2).GE.5)GO TO 45
      IF(SUM.LE.1.33*SUMMAX)GO TO 40
   45 SUMMAX=SUM
      IMAX=I
   40 CONTINUE
      X=IMAX
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE HARM(X.XF)
C----CALCULATE NEW SPECTRUM(F) FROM OLD SPECTRUM(IN A). PITCH(FROM X).
C    AND FACTOR BY WHICH NON-HARMONIC LINES ARE TO BE MULTIPLIED (XF)
      COMMON /BUFFER2/ F(512.2).G(512.2).A(512.2)
      DO 80 I=1.512
      F(I.1)=A(I.1)*XF
```

```
80    F(I,2)=A(I,2)*XF
      DO 85 I=154,360
      F(I,1)=0.
85    F(I,2)=0.
      IMAX=X+.5
      DO 90 I=IMAX,3000,IMAX
      XI=I
      LNUM=XI/19.53125+.5
      LNUM=LNUM+1
      F(LNUM,1)=A(LNUM,1)
      F(LNUM,2)=A(LNUM,2)
      F(514-LNUM,1)=A(514-LNUM,1)
90    F(514-LNUM,2)=A(514-LNUM,2)
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE TAPDAT(J)
C----TRANSLATE IOBUF(I),I=1,512 INTO IDATA(I), IST OF 2ND HALF
C     IOBUF IS IN TAPE BINARY FORMAT. IDATA IS IN NUMERICAL FORMAT
      COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
      K=(J-1)*1024
      DO 20 I=1,512
      IDATA(2*I-1+K)=0
      IDATA(2*I+K)=0
      FLD(21,11,IDATA(2*I-1+K))=FLD(1,11,IOBUF(I))
      FLD(32,4,IDATA(2*I-1+K))=FLD(14,4,IOBUF(I))
      FLD(21,11,IDATA(2*I+K))=FLD(19,11,IOBUF(I))
      FLD(32,4,IDATA(2*I+K))=FLD(32,4,IOBUF(I))
      IF(FLD(0,1,IOBUF(I)).EQ.0)GO TO 10
      IDATA(2*I-1+K)=-IDATA(2*I-1+K)
10    IF(FLD(18,1,IOBUF(I)).EQ.0)GO TO 20
      IDATA(2*I+K)=-IDATA(2*I+K)
20    CONTINUE
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE DATTAP
C----TRANSLATE JODATA(I) INTO IOBUF(I)
```

```
      COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
      DO 100 I=1,512
      IOBUF(I)=0
      IF(JODATA(2*I-1).GE.0)GO TO 10
      JODATA(2*I-1)=-JODATA(2*I-1)
      FLD(0,1,IOBUF(I))=1
10    FLD(1,11,IOBUF(I))=FLD(21,11,JODATA(2*I-1))
      FLD(14,4,IOBUF(I))=FLD(32,4,JODATA(2*I-1))
      IF(JODATA(2*I).GE.0)GO TO 20
      JODATA(2*I)=-JODATA(2*I)
      FLD(18,1,IOBUF(I))=1
20    FLD(19,11,IOBUF(I))=FLD(21,11,JODATA(2*I))
100   FLD(32,4,IOBUF(I))=FLD(32,4,JODATA(2*I))
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE WEIGHT3
C----MODIFY F BY CURVE OF APPROXIMATE RELATIVE IMPORTANCE FOR INTELLIGIBILITY
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 10 I=1,51
      XI=I
      F(I,1)=F(I,1)*(XI*19.53125)**3/1.E09
10    F(I,2)=F(I,2)*(XI*19.53125)**3/1.E09
      DO 20 I=102,257
      XI=I
      F(I,1)=F(I,1)*4.E06/(XI*19.53125)**2
20    F(I,2)=F(I,2)*4.E06/(XI*19.53125)**2
      DO 30 I=2,256
      F(514-I,1)=F(I,1)
30    F(514-I,2)=F(I,2)
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE NORM1(I,JJ,SFACT)
C----CALCULATE SFACT- THE NORMALIZATION FACTOR BASED UPON 40 $ 380
      COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      OMAX=1.E-37
      DO 10 II=1,512
```

```
10   OMAX=AMAX1(OMAX.ABS(F(II,1)))
     IF((I.EQ.1).AND.(JJ.EQ.1))GO TO 20
     IF(OMAX*SFACT.GT.380.)GO TO 30
     IF(OMAX*SFACT.LT.40.)GO TO 40
     RETURN
20   SFACT=40./OMAX
     RETURN
30   SFACT=SFACT*.6
     RETURN
40   SFACT=SFACT*1.2
     RETURN
     END
C-----------------------------------------------------------------------
     SUBROUTINE NORM2(I.JJ.N.SFACT)
C----CALCULATE SFACT BY UPDATING PREVIOUS VALUE BY .5*(RATIO-SFACT)
     COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY.IDATA(2048).JODATA(1280)
     COMMON /BUFFER2/ F(512.2),G(512.2).A(512.2)
     OMAX=1.E-37
     DO 10 II=1,256
     XII=II
     VAL=(AMAX1(F(II,1).F(513-II,1)))*((257.-XII)/XII)
10   OMAX=AMAX1(OMAX.VAL)
     INMAX=0
     DO 20 II=N,N+511
20   INMAX=MAXO(INMAX.IDATA(II))
     XINMAX=INMAX
     RATIO=XINMAX/OMAX
     IF((I.EQ.1).AND.(JJ.EQ.1))GO TO 30
     SFACT=SFACT+.5*(RATIO-SFACT)
     RETURN
30   SFACT=RATIO
     RETURN
     END
C-----------------------------------------------------------------------
C----TEMP TEST
     SUBROUTINE TEST1
     COMMON /BUFFER2/ F(512.2),G(512.2).A(512.2)
     CALL TTW
```

```
      CALL FFT
      CALL IFFT
      RETURN
      END
      SUBROUTINE TEST2
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      CALL TTW
      CALL FFT
      CALL FFT
      CALL IFFT
      CALL IFFT
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE TABLE(N)
C----GENERATE SIN AND COS TABLE FOR FFT & IFFT
      COMMON /SCTAB/ S(512)
      DO 100 I=1,10
      K=2**I
100   IF(N.EQ.K)GO TO 101
      STOP
C-----ERROR THAT N IS NOT MOD 2 OR N IS GREATER THAN 1024 OR LESS THAN 2
101   K=I
      L=2**(K-1)
      PI=3.14159
      DO 103 J=1,L
      P=PI*FLOAT(J-1)/FLOAT(L)
      S(J)=SIN(P)
103   CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE ZLOW2
C----SUPPRESS LOW TEN ELEMENTS TO EQUIL TWICE FORMANT PEAK
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      FMAX=SORT(F(1,1)**2+F(1,2)**2)
      DO 100 I=2,221
100   FMAX=AMAX1(FMAX,SORT(F(I,1)**2+F(I,2)**2))
```

```
      FACT=SQRT(F(257,1)**2+F(257,2)**2)/FMAX
      DO 110 I=248,266
      F(I,1)=F(I,1)*2/FACT
  110 F(I,2)=F(I,2)*2/FACT
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE ABSVF
C----TAKE ABSOLUTE VALUE OF ARRAY F
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      DO 100 I=1,512
  100 F(I,1)=ABS(F(I,1))
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE NOISUB2(XNOISE)
C----SUBTRACT 3*XNOISE FROM MAG(F)
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      XNOISE=XNOISE*3.
      DO 30 I=1,512
      XMAG=SQRT(F(I,1)**2+F(I,2)**2)
      XMAG=XMAG-XNOISE
      IF(XMAG.GE.0.)GO TO 40
      F(I,1)=0.
      F(I,2)=0.
      GO TO 30
   40 F1=XMAG*COS(ATAN2(F(I,2),F(I,1)))
      F(I,2)=XMAG*SIN(ATAN2(F(I,2),F(I,1)))
      F(I,1)=F1
   30 CONTINUE
      RETURN
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE NPAP
C----POWER SPECTRAL OPTIMIZATION WITH PAST NOISE WEIGHTING AND SPEECH/
C    NO-SPEECH TEST
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      COMMON /START/ INIT(16),RUNWGT(32)
      CALL TTW
      CALL FFT
      CALL ESTNO2(XNOISE)
      CALL WEIGH3(XNOISE)
      CALL IFFT
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE WEIGH3(XNOISE)
C----MODIFY POWER SPECTRUM AS IN WEIGH2 BUT WITH ESTIMATED NOISE FROM
C    ESTNOISE2
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
C----CALCULATE ARRAY G. THE LONG TERM AVERAGE FOR NORM. SPEECH
      G(1,1)=0.:G(2,1)=1./2.**8:G(3,1)=1./2.**5
      G(4,1)=1./12.:G(5,1)=.5
      DO 90 I=6,27
90    G(I,1)=1.
      DO 100 I=28,257
      XI=I
      FREQ=(XI-1.)*19.53125
100   G(I,1)=(500./FREQ)**2
      SUM=0.
      DO 110 I=1,257
110   SUM=SUM+G(I,1)
      SUM=SUM/257.
C----CALCULATE AVERAGE OF MAG(F)
      FSUM=0.
      DO 115 I=1,257
115   FSUM=FSUM+SQRT(F(I,1)**2+F(I,2)**2)
      FSUM=FSUM/257.
      DO 130 I=1,257
130   G(I,1)=G(I,1)*FSUM/SUM
```

```
      DO 140 I=2,256
  140  G(514-I,1)=G(I,1)
      DO 150 I=1,512
      RE=F(I,1)
      XIM=F(I,2)
      XMAG2=F(I,1)**2+F(I,2)**2
      XMAG=(G(I,1)**2*XMAG2)/(G(I,1)**2+XNOISE**2)
      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
  150 F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE ESTNO2(XNOISE)
C----ESTIMATE AVERAGE NOISE LEVEL FROM 50 PAST SAMPLES
      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
      COMMON /START/ INIT(16),RUNWGT(32)
C----CALCULATE AVERAGE SPECTRUM FOR EACH FRAME
      SUMA=0.
      DO 10 I=2,257
   10 SUMA=SUMA+SQRT(F(I,1)**2+F(I,2)**2)
      AVER=SUMA/256.
C----CALCULATE VARIANCE FOR EACH FRAME
      SUMV=0.
      DO 11 I=2,257
   11 SUMV=SUMV+(SQRT(F(I,1)**2+F(I,2)**2)-SUMA)**2
      VAR=SQRT(SUMV)/256.
C----INIT(1) IS FRAME COUNT TO 50
      IF(INIT(1).NE.1)GO TO 15
      INIT(1)=INIT(1)+1
C----RUNWGT(1) IS AVERAGE SUMA FOR SPEECH
C----RUNWGT(2) IS AVERAGE SUMA FOR NOISE
C----INIT(2) IS 1-SPEECH FRAME, 0-NOISE FRAME
      RUNWGT(1)=SUMA
      RUNWGT(2)=SUMA-.1*SUMA
      INIT(2)=1
      XNOISE=RUNWGT(2)
C----RUNWGT(3) IS AVERAGE SPEECH VARIANCE
C----RUNWGT(4) IS AVERAGE NOISE VARIANCE
```

```
C----RUNWGT(5) IS THRESHOLD FOR NOISE-SPEECH
      RUNWGT(3)=SUMV
      RUNWGT(4)=SUMV
      RUNWGT(5)=.92*SUMA
      RETURN
C----UPDATE RUNNING AVERAGES FOR THRESHOLD
 15   IF(INIT(1).GT.50)GO TO 12
      INIT(1)=INIT(1)+1
 12   IF(SUMA.GT.RUNWGT(5))GO TO 13
      K1=2
      K2=4
      INIT(2)=0
      GO TO 14
 13   K1=1
      K2=3
      INIT(2)=1
 14   FINIT=FLOAT(INIT(1))
      RUNWGT(K1)=(FINIT_1.)*RUNWGT(K1)/FINIT+SUMA/FINIT
      RUNWGT(K2)=(FINIT-1.)*RUNWGT(K2)/FINIT+SUMV/FINIT
      RATIO=RUNWGT(1)/(RUNWGT(1)+RUNWGT(2))
      RUNWGT(5)=RUNWGT(1)-1.2*(RUNWGT(1)-RUNWGT(2))*RATIO
      XNOISE=RUNWGT(2)
      RETURN
      END

ready
```

ATE
LMED